

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

DAVID VITOR ANTONIO

IMPLEMENTAÇÃO DE PROTÓTIPO BASEADO NA TECNOLOGIA OCR
APLICADA AO RECONHECIMENTO DE RÓTULOS PARA BUSCA EM BANCO
DE DADOS

CRICIUMA
2019

DAVID VITOR ANTONIO

**IMPLEMENTAÇÃO DE PROTÓTIPO BASEADO NA TECNOLOGIA OCR
APLICADA AO RECONHECIMENTO DE RÓTULOS PARA BUSCA EM BANCO
DE DADOS**

Trabalho de Conclusão de Curso, apresentado
para obtenção do grau de Bacharel no curso de
Ciência da Computação da Universidade do
Extremo Sul Catarinense, UNESC.

Orientador: Prof. Me. Gustavo Bisognin

CRICIUMA

2019

DAVID VITOR ANTONIO

**IMPLEMENTAÇÃO DE PROTÓTIPO BASEADO NA TECNOLOGIA OCR
APLICADA AO RECONHECIMENTO DE RÓTULOS PARA BUSCA EM BANCO
DE DADOS**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Projeto de Software.

Criciúma, 27 de junho de 2019.

BANCA EXAMINADORA



Prof. Gustavo Bisognin - Mestre - (UNESC) - Orientador



Prof. Gilberto Vieira da Silva - Especialista - (UNESC)



Prof. Anderson Rodrigo Farias - Especialista - (SATC/UNIBAVE)

Dedico este trabalho aos meus pais, que me apoiaram desde o início desta jornada.

AGRADECIMENTOS

Agradeço aos meus pais, Sergio e Edna, que me apoiaram desde o início dessa trajetória, que mesmo em meio a dificuldades, sempre encontraram um meio de me ajudar com o que precisava.

Agradeço também a minha namorada, que esteve ao meu lado durante o desenvolvimento deste trabalho, sendo sempre compreensiva e me apoiando.

Agradeço ao professor e orientador Gustavo, que me incentivou e ajudou a concluir este trabalho, sempre apontando melhorias, indicando erros e parabenizando pelos acertos. Além disso, agradeço a todos os professores do curso, estes que me ajudaram a crescer como profissional, estudante e pessoa.

“O conformismo é carcereiro da liberdade e o inimigo do crescimento.”

John Kennedy

RESUMO

A Internet das coisas visa conectar todos as coisas presentes no dia a dia para a facilitar e automatizar a rotina humana. Desde celulares; guarda chuvas; geladeiras e diversos outros itens, cada vez mais pode se observar objetos sendo conectados a uma rede e realizando a comunicação com outros dispositivos. Porém, coisas degradáveis como embalagens de produtos também podem fazer deste conjunto de itens detectáveis, dessa forma, seria possível por exemplo, visualizar informações mais detalhadas sobre um produto no celular apenas ao aproximar o dispositivo com o rótulo. Porém, atualmente, o custo para tornar todos os rótulos detectáveis pode tornar inviável a detecção de produtos descartáveis. Dessa forma, como alternativa ao problema descrito, este trabalho apresenta a implementação de um protótipo de arquitetura e software para que seja realizada a detecção dos rótulos através da câmera do dispositivo utilizado, isso é possível através do reconhecimento ótico de caracteres (OCR), este que pode ser realizado através de uma plataforma de visão computacional já existente. O protótipo pode ser acessado via navegador de internet de um computador, dessa forma, ao acessá-lo e capturar uma foto do rótulo de uma cerveja, a foto é enviado para o servidor, este que realiza o OCR através da plataforma de visão computacional, busca as informações em um banco de dados referentes a cerveja indicada e mostra para o usuário através do navegador, permitindo também que o usuário avalie a cerveja. Além disso, durante o trabalho são realizados testes com redes neurais convolucionais para a avaliação de viabilidade para utilizá-las em um mecanismo de detecção com classificadores, para isto, foram estudados os conceitos de redes neurais e suas arquiteturas, além da implementação de um pequeno protótipo rede neural convolucional (CNN) para teste. Dessa forma, comprovou-se inviabilidade de utilização de classificadores de imagens já que na arquitetura proposta o usuário seria o responsável por inserir novas informações e o treinamento de classificadores com CNN requer uma quantidade significativa de dados de imagens.

Palavras-chave: Internet das coisas. OCR. Visão Computacional. Redes Neurais. Rótulos.

ABSTRACT

Internet of Things is a concept that aims at connecting all the things that are used in the daily routine to automate and make human duties easier. From Smartphones to umbrellas; refrigerators and other items. It can be seen that the number of objects being connected to a network and communicating with other devices is increasing. However, disposable things like products' packages can also become part of this list of detectable things, thus, it would be possible, for instance, to visualize detailed information about a product using smartphones only by approximating the device with the product's label. However, nowadays, the cost of turning a label detectable can make the product detection inviable. Thus, as an alternative to the described problem, this paper shows an implementation of an architecture and prototype that uses optical character recognition (OCR) to do label recognition through the use of an existing computer vision platform. The prototype can be accessed using the web browser of a computer with internet access, so if the user accesses the app and sends a picture of a beer label, the image is sent to a server that communicates with the computer vision platform, later, the returned information will be used to find information about the beer in the database and it will be sent to the user, also allowing the user to rate the product. Furthermore, tests to check if the usage of Convolutional Neural Networks (CNN) as classifier can meet the needs of detection mechanism were performed, that was possible while studying concepts of neural networks, its architecture and also developing a small CNN. In such a way, it was possible to confirm that it would not be possible to use image classifiers since that in the proposed architecture the user would be responsible for submitting new data and it requires a significant amount of image data to train a CNN.

Keywords: Internet of Things. OCR. Computer Vision. Neural Networks. Labels.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - Estrutura básica do código HTML | 10 |
| Figura 2 - Navegador interpretando o código HTML | 11 |
| Figura 3 - Formato JSON para armazenamento de informações em texto | 14 |
| Figura 4 - Visão computacional aplicada a carros autônomos | 17 |
| Figura 5 - Modelo de rede neural <i>feedforward</i> | 19 |
| Figura 6 - Modelo de neurônio Biológico..... | 20 |
| Figura 7 - Modelo do neurônio artificial | 21 |
| Figura 8 - Rede neural Perceptron | 23 |
| Figura 9 - Perceptron Multicamadas | 24 |
| Figura 10 - Perceptron com realimentação | 24 |
| Figura 11 - Exemplo de Rede Kohonen | 25 |
| Figura 12 – Operação convolucional em grade de uma dimensão | 27 |
| Figura 13 – Rede neural convolucional para processamento de imagens de tamanho 28x28 | 27 |
| Figura 14 - Reconhecimento de formas das letras..... | 31 |
| Figura 15 - Google Tradutor, tradução em tempo real | 32 |
| Figura 16 - Troca de mensagens usando XML | 36 |
| Figura 17 - Modelos de camadas de IoT | 38 |
| Figura 18 – Modelo de CNN..... | 46 |
| Figura 19 – Cervejas para testes em plataformas..... | 48 |
| Figura 20 – Variações de imagem para testes em plataformas de CV | 48 |
| Figura 21 – Diagrama de arquitetura do núcleo do protótipo | 50 |
| Figura 22 – Diagrama de interação do usuário do aplicativo web | 51 |
| Figura 23 – Gráfico de quantidade de downloads das principais frameworks <i>frontend web</i> | 52 |
| Figura 24 – Criação de componentes em ReactJS | 53 |
| Figura 25 – Quatro rótulos para treinamento de CNN..... | 54 |
| Figura 26 – Resultado de testes de CNN com baixo número de exemplos | 55 |
| Figura 27 – Resultado de teste de CNN com alto número de exemplos..... | 55 |
| Figura 28 – Gráfico de precisão geral na comparação de plataforma de CV | 58 |
| Figura 29 – Gráfico de precisão por situação na comparação de plataforma de CV | 59 |
| Figura 30 – Tela de login do protótipo | 60 |

| | |
|---|----|
| Figura 31 – Tela de catálogo e reconhecimento de rótulo do protótipo..... | 60 |
| Figura 32 – Tela de cadastro de cerveja do protótipo | 61 |

LISTA DE TABELAS

| | |
|--|----|
| TABELA 1 – PALAVRAS IMPORTANTES PARA COMPARAÇÃO DE PLATAFORMAS DE CV | 49 |
| TABELA 2 – RESULTADO DE TESTES ENTRE PLATAFORMAS DE CV | 56 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|--|
| API | <i>Application Programming Interface</i> |
| CV | <i>Computer Vision</i> |
| HTML | <i>HyperText Markup Language</i> |
| HTTP | <i>HyperText Transfer Protocol</i> |
| IA | Inteligência Artificial |
| IoT | <i>Internet of Things</i> |
| JSON | <i>Javascript Object Notation</i> |
| OCR | <i>Optical Character Recognition</i> |
| XML | <i>Extensible Markup Language</i> |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 6 |
| 1.1 OBJETIVO GERAL | 7 |
| 1.2 OBJETIVOS ESPECÍFICOS | 7 |
| 1.3 JUSTIFICATIVA | 7 |
| 1.4 ESTRUTURA DO TRABALHO | 8 |
| 2 MODELOS DE EXTRAÇÃO DE DADOS | 10 |
| 2.1 SPIDERING..... | 12 |
| 2.2 WEB SCRAPING | 12 |
| 2.2.1 Bibliotecas | 13 |
| 2.2.2 Frameworks | 14 |
| 2.2.3 Aplicações | 15 |
| 3 INTELIGÊNCIA ARTIFICIAL APLICADA AO RECONHECIMENTO DE IMAGENS | 17 |
| 3.1 SISTEMAS DE CLASSIFICAÇÃO DE IMAGENS | 19 |
| 3.1.1 Redes neurais artificiais | 19 |
| 3.1.1.1 Neurônio Artificial | 20 |
| 3.1.1.2 Arquiteturas de redes neurais | 22 |
| 3.1.1.2.1 <i>Arquitetura feedfoward de camada simples</i> | 22 |
| 3.1.1.2.2 <i>Arquitetura feedfoward de camadas múltiplas</i> | 23 |
| 3.1.1.2.3 <i>Arquitetura recorrente</i> | 24 |
| 3.1.1.2.4 <i>Arquitetura com estrutura reticulada</i> | 25 |
| 3.1.1.3 Redes Neurais Convolucionais | 25 |
| 3.1.1.4 Treinamento de redes neurais..... | 28 |
| 4 RECONHECIMENTO ÓTICO DE CARACTERES..... | 30 |
| 4.1 IBM WATSON | 32 |
| 4.1.1 Reconhecimento de texto com IBM Watson | 33 |
| 4.2 WINDOWS AZURE | 33 |
| 4.2.1 Reconhecimento de texto com Windows Azure | 33 |
| 4.3 AMAZON REKOGNITION | 34 |
| 4.3.1 Reconhecimento de texto com Amazon Rekognition | 34 |
| 4.4 GOOGLE CLOUD VISION | 34 |
| 4.4.1 Reconhecimento de texto com Google Cloud Vision | 34 |

| | |
|---|-----------|
| 5 WEB API..... | 36 |
| 6 INTERNET DAS COISAS | 38 |
| 6.1 CAMADA DE OBJETO..... | 39 |
| 6.2 CAMADA DE ABSTRAÇÃO | 39 |
| 6.3 CAMADA DE GERENCIAMENTO DE SERVIÇO | 39 |
| 6.4 CAMADA DE APLICAÇÃO..... | 40 |
| 6.5 CAMADA DE NEGÓCIO | 40 |
| 7 TRABALHOS CORRELATOS..... | 41 |
| 7.1 PROTÓTIPO DE APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM IMAGENS PARA BUSCA SEMÂNTICA SOBRE RÓTULOS DE CERVEJAS | 41 |
| 7.2 PROPOSTA DE ARQUITETURA DE UM SISTEMA COM BASE EM OCR NEURONAL PARA RESGATE E INDEXAÇÃO DE ESCRITAS PALEOGRÁFICAS DO SEC.XVI AO XIX | 42 |
| 7.3 RECONHECIMENTO DE DÍGITOS EM IMAGENS DE MEDIDORES DE CONSUMO DE GÁS NATURAL UTILIZANDO TÉCNICAS DE VISÃO COMPUTACIONAL | 42 |
| 7.4 SISTEMA DE DIGITALIZAÇÃO, DETECÇÃO E CLASSIFICAÇÃO AUTOMÁTICA DE RECIBOS EM DISPOSITIVOS MÓVEIS..... | 43 |
| 8 IMPLEMENTAÇÃO DE PROTÓTIPO BASEADO NA TECNOLOGIA OCR APLICADA AO RECONHECIMENTO DE RÓTULOS PARA BUSCA EM BANCO DE DADOS | 44 |
| 8.1 METODOLOGIA..... | 44 |
| 8.1.1 Métodos alternativos ao OCR para detecção de rótulos | 45 |
| 8.1.2 Comparações entre plataformas de visão computacional | 47 |
| 8.1.2.1 Metodologia para comparação de plataformas de visão computacional | 47 |
| 8.1.3 Estrutura do protótipo da API | 49 |
| 8.1.4 Arquitetura do núcleo do protótipo | 50 |
| 8.1.5 Arquitetura da aplicação web do protótipo..... | 51 |
| 8.1.5.1 ReactJS..... | 52 |
| 8.2 RESULTADOS | 53 |
| 8.2.1 Métodos alternativos ao OCR para reconhecimento de rótulos | 54 |
| 8.2.2 Comparações entre plataformas de visão computacional | 56 |
| 8.2.3 Aplicação | 59 |
| 8.2.4 Extração de dados para composição de banco | 62 |

| | |
|--------------------------|-----------|
| 9 CONCLUSÃO | 64 |
| REFERÊNCIAS..... | 66 |
| APENDICE(S)..... | 71 |
| ANEXO(S)..... | 72 |

1 INTRODUÇÃO

A “Internet das Coisas” se refere a uma revolução tecnológica que tem como objetivo conectar os itens usados do dia a dia à rede mundial de computadores, “[...] a IoT representa a próxima evolução da Internet, dando um grande salto na capacidade de coletar, analisar e distribuir dados que nós podemos transformar em informações, conhecimento e, por fim, sabedoria” (EVANS, 2011, p.2). Neste contexto, cada vez mais, os dispositivos computacionais se conectam e se comunicam com as coisas do nosso cotidiano. Atualmente se pode observar o surgimento de diversos eletrônicos como eletrodomésticos e meios de transporte que estão conectados à Internet e a outros dispositivos, como computadores e smartphones.

Porém, para a possível quebra das barreiras tecnológicas que geralmente limitam a IoT preferencialmente para dispositivos eletrônicos, é necessário unir as mais diversas áreas da tecnologia, como por exemplo, o campo da Inteligência Artificial (IA). A IA apresentou um crescimento significativo nos últimos anos, proporcionando o desenvolvimento de diversos tipos de algoritmos para realizar processamento similar ao raciocínio humano. Dessa forma, foram encontradas possibilidades de aplicações do raciocínio artificial através da Visão Computacional para permitir a extração de características de imagens através de algoritmos avançados. Com isso, um sistema de reconhecimento de imagens associado a um banco de dados com informações sobre produtos, poderá realizar uma análise das imagens de rótulos de produtos descartáveis e disponibilizar os dados para o consumidor. Diante dos fatos apresentados, este trabalho propõe a utilização da funcionalidade de reconhecimento de caracteres de imagens presente em plataformas de visão computacional para a criação de um protótipo que poderá realizar a detecção de cervejas através de fotos dos seus rótulos. Assim, o protótipo é capaz de utilizar a cerveja detectada para buscar informações sobre o produto em um banco de dados interno que possui informações sobre o produto. Além disso, para a escolha da plataforma de visão computacional ideal, o trabalho realiza uma comparação de desempenho entre 4 plataformas de visão computacional disponíveis para a utilização.

1.1 OBJETIVO GERAL

Disponibilizar uma estrutura de interface para aplicações que permita reconhecimento e detalhamento de rótulos utilizando banco de dados e OCR através de visão computacional.

1.2 OBJETIVOS ESPECÍFICOS

Os seguintes objetivos específicos serão utilizados para o desenvolvimento do trabalho:

- a) realizar a aquisição de informações dos produtos para composição de banco de dados;
- b) efetuar a comparação e testes de plataformas de visão computacional para avaliar desempenho no reconhecimento de caracteres;
- c) implementar um protótipo para a integração da leitura de caracteres de rótulos com busca no banco de dados para detalhamento de produto;
- d) desenvolver uma interface para permitir integração do protótipo obtido.

1.3 JUSTIFICATIVA

Pode-se identificar no mercado atual, a existência de algumas aplicações baseadas no Reconhecimento Ótico de Caracteres, do inglês *Optical Character Recognition* (OCR), estas que são aplicadas nas mais variadas áreas da indústria. Desta forma, é correto afirmar que a tecnologia OCR pode apoiar soluções de acesso à informação de forma fácil e rápida, bastando acessar um aplicativo no celular ou outro dispositivo conectado a internet que tenha uma câmera acoplada.

A constante evolução da tecnologia da informação associado ao conceito de Internet das Coisas tem alterado a forma como as coisas e a sociedade atual é organizada. Uma das áreas que mais cresce atualmente no Brasil, é a produção de cervejas artesanais, a qual apresenta diversas formulações e também novos rótulos, os quais, muitas vezes, não possui a devida licença requerida pelo Ministério da Agricultura e Pecuária (MAPA). Desta forma, é bastante difícil conhecer as diversas

variações de produtos e formulações, uma vez que, estes produtos, não possuem muita especificação técnica em seus rótulos.

Diante disso, existe a necessidade de um estudo e desenvolvimento de uma API para leitura e detalhamento de rótulos de forma dinâmica. Além disso, a implantação do *software* em nuvem e disponibilização como serviço em uma estrutura API permitirá que a aplicação possa ser utilizada entre os mais diversos produtos. De acordo com o que foi acima apresentado, o desenvolvimento de um sistema baseado na tecnologia de OCR aplicada ao reconhecimento de rótulos se justifica.

1.4 ESTRUTURA DO TRABALHO

Este projeto é dividido em oito capítulos, onde o primeiro introduz os conceitos e informações que levaram ao desenvolvimento deste projeto de pesquisa, assim como também define os objetivos para o alcance do resultado desejado e sua justificativa.

O segundo capítulo irá abordar as formas de extração de dados da internet, dessa forma, o objetivo de extrair informações sobre produtos para composição de um banco de dados detalhado poderá ser alcançado.

O terceiro capítulo detalhará as tecnologias disponíveis para a realização de análise de imagens, permitindo assim a leitura de rótulos de produtos, e que neste trabalho tem o objetivo de ler rótulos de cervejas. Dentro os conceitos abordados de tecnologias que permitem a análise de imagens está a rede neural, esta que possuirá um detalhamento avançado pois irá compor as tecnologias utilizadas para o trabalho proposto.

O quarto capítulo abordará a história e estado atual de tecnologias que permitem a realização de OCR. Além disso, serão apresentadas algumas plataformas de visão computacional comerciais que permitem a realização de OCR via API.

No quinto capítulo será explicado o conceito de API, o que são e como são disponibilizadas, este possui o objetivo de esclarecer a arquitetura do protótipo, e também, demonstrar como o protótipo poderá ser utilizado como serviço de reconhecimento de produtos.

O sexto explicará um pouco sobre o conceito de internet das coisas e de que maneira este conceito foi utilizado como filosofia do trabalho proposto.

O sétimo capítulo abordará alguns trabalhos acadêmicos realizados com o objetivo similares ao abordado nesse trabalho.

O oitavo capítulo apresentará a metodologia utilizada para o alcance do objetivo do trabalho, além de apresentar os resultados obtidos.

2 MODELOS DE EXTRAÇÃO DE DADOS

Um protótipo que realiza a busca informações sobre produtos em uma base de dados interna necessita de informações para compor a sua base de conhecimento. Dessa forma, a internet possui uma grande quantidade de informações que cresce cada dia mais, porém, a maioria dessas informações são difíceis de extrair das páginas devido à falta de estrutura de seu conteúdo, porém, ainda assim existem diversas páginas bem estruturadas que em conjunto com técnicas de extração de informações da web, é uma rica fonte de informações (ARASU; GARCIA-MOLINA, 2003, tradução nossa).

A internet moderna utiliza para compor as suas páginas a linguagem *Hyper Text Markup Language* (HTML), esta que desde o surgimento dos navegadores vem recebendo atualizações para se adequar as necessidades do período. O HTML é uma linguagem de marcação que utiliza de marcadores, mais conhecidos como *tags*. Desta forma, conforme pode ser observado na figura 1, a estrutura principal da linguagem é composta de *tags* semânticas e funcionais, onde os elementos que compõe o código, chamados de elementos HTML, são geralmente conteúdo em texto entre uma *tag* de início e outra de fim (DUCKETT, 2011, tradução nossa).

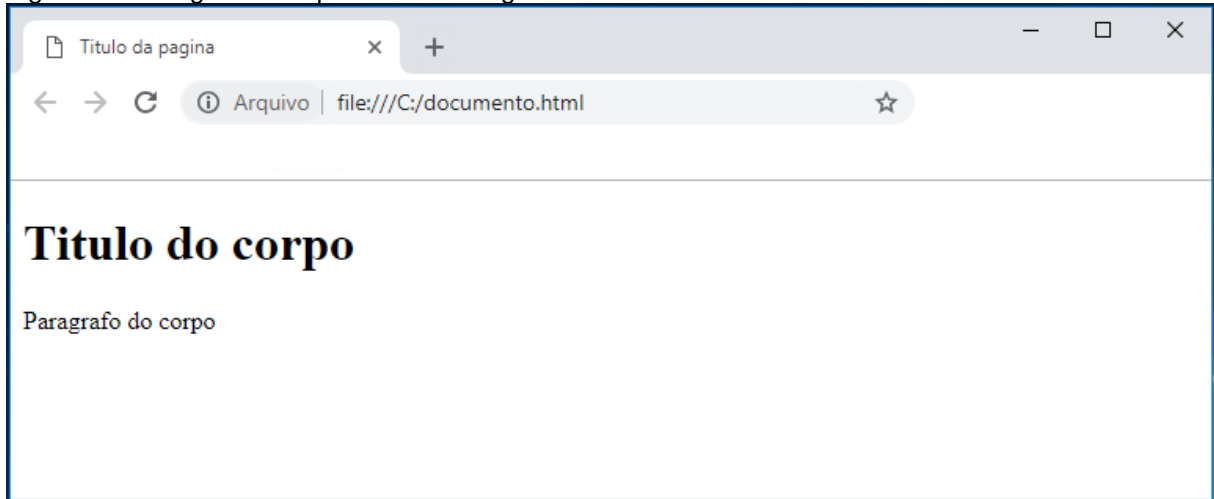
Figura 1 - Estrutura básica do código HTML



Fonte: Do autor.

O documento com o código é então lido e interpretado pelo navegador utilizado para abrir o documento, e conforme pode ser observado na figura 2, este realiza a renderização para que o usuário que visualize o conteúdo.

Figura 2 - Navegador interpretando o código HTML



Fonte: Do autor.

Os conjuntos de *tags* de início e fim funcionam como containers, categorizando o texto que guardam, permitindo uma leitura semântica (DUCKETT, 2011, tradução nossa).

A partir da interpretação do código HTML, é possível realizar leituras das *tags* e seus atributos, permitindo assim serem extraídas apenas as informações desejadas em uma consulta aos documentos que são baixados pelo navegador quando acessados os sites modernos.

Para extração automática das informações presentes nas páginas da internet são utilizados robôs baseados em *software*. Robôs são dispositivos que realizam tarefas repetitivas, similar a uma linha de montagem, dessa forma, qualquer programa que realiza um acesso à internet e extrai informações pode ser chamado de robô. (HEATON, 2002, tradução nossa)

Existem algumas diferentes maneiras de se aplicar um robô na leitura de informações das páginas, as principais são através das técnicas de *spidering* e *web scraping*. Além disso, é possível buscar informações presentes em bancos de dados hospedados em servidor na internet ou outra aplicação web através de chamadas a uma Web API.

2.1 SPIDERING

Para a técnica de *spidering*, é utilizado um robô especializado chamado de Aranha, do inglês *Spider*, ou *Web Crawler*. O robô tem o objetivo de vasculhar uma página HTML inicial, a qual lhe foi fornecida, em busca de informações e referências para outras páginas HTML, após, irá acessar as páginas referenciadas e realizar o mesmo procedimento. Dessa forma, o *Web Crawler* poderá percorrer uma quantidade significativa de páginas para obter as informações desejadas, a quantidade irá depender dos recursos e limites estipulados durante o seu desenvolvimento (HEATON, 2002, tradução nossa).

Além disso, o *Web Crawler* pode ser também utilizado para mapeamento de sites, procurando a partir da página inicial as imagens, arquivos e referências para fornecer uma visão geral da estrutura do site. Dessa forma, é possível até mesmo, identificar problemas na estrutura do site analisado, como por exemplo, encontrar links inválidos presentes nos documentos HTML que o compõe (HEATON, 2002, tradução nossa).

2.2 WEB SCRAPING

Web Scraping, também conhecida por *Screen Scraping*, é a técnica que consiste em realizar a leitura de páginas da web previamente configuradas para a extração de dados, de forma em que se possa gerar uma saída estruturada para leitura, análise ou processamento (CORDING, 2018, tradução nossa).

O processo de extração de informações é em sua maioria realizado utilizando robôs programados para leitura de uma estrutura HTML específica, dessa forma, cada documento deverá ter um robô desenvolvido especificamente para a sua estrutura.

As técnicas para *web scraping* requerem uma quantidade significativa de tecnologias e técnicas para o seu funcionamento, como análise de dados e segurança da informação (MITCHELL, 2015, tradução nossa). Existem algumas maneiras de realizar a implementação de um robô de *Web Scraping*, elas podem ser divididas em

3 categorias principais: Bibliotecas, *Frameworks* e Aplicações (GLEZ-PEÑA et. al., 2014, tradução nossa).

2.2.1 Bibliotecas

A utilização de bibliotecas permite a maior flexibilidade na implementação de um robô de *Data Scraping*, uma vez que ela poderá ser realizada utilizando qualquer linguagem de programação desejada. Dessa forma, geralmente para implementação do robô seria incluída uma biblioteca para conexão HTTP, esta que irá permitir a conexão com o site desejado.

Uma das bibliotecas mais famosas para acesso de site é a Curl, ela possui diversas funcionalidades do protocolo HTTP já inclusas, como acesso com certificado SSL, HTTP POST, HTTP PUT, Upload FTP, Upload HTML, Proxies, Cookies e autenticação HTTP. Além disso, ela possui diversas integrações com as mais diversas linguagens de programação (GLEZ-PEÑA et. al., 2014, tradução nossa).

Além de realizar a conexão com o site e baixar o conteúdo HTML da página, será necessário interpretar o código da página obtido para extração dos dados desejados, tarefa que pode ser feita utilizando um processador de texto como AWK.

A versão original da linguagem AWK foi escrita em 1977, ela é uma linguagem para automação no processamento de textos, dessa forma, por exemplo, é possível através dos scripts AWK ler e descartar certas linhas de um texto baseado em condições ou padrões. Na linguagem também é possível realizar diversas outras tarefas, como armazenamento em memória, porém, a sua sintaxe foi desenvolvida para tornar fácil a manipulação de textos (GNU.ORG, 2018, tradução nossa).

Outra alternativa para leitura do conteúdo seria uma biblioteca para realizar o *parsing* do código HTML, como por exemplo “PHP Simple HTML Dom”. Esta biblioteca para PHP realiza a leitura do código HTML e retorna objetos com métodos similares aos existentes nos navegadores para manipulação da Arvore de elementos (DOM), dessa forma, é possível realizar buscas nas tags da página programaticamente e extrair as informações necessárias.

Após a extração de dados, eles deverão ser organizados de forma que possam ser lidos ou processados por outros programas, algumas das estruturas mais

famosas que podem ser utilizadas para armazenamento da informação em formato texto são XML e JSON.

O *Javascript Object Notation* (JSON), conforme pode ser observado na figura 3, é um formato para armazenamento de informações em formato texto que surgiu da linguagem Javascript, porém, atualmente é utilizado nas mais diversas linguagens, que suportam o formato nativamente ou através de bibliotecas.

Figura 3 - Formato JSON para armazenamento de informações em texto



Fonte: Do autor.

Além disso, outro possível destino para os dados extraídos seria um banco de dados estruturado como MySQL ou Oracle, dessa forma, para alcançar esse objetivo existem dois principais métodos: o robô de extração poderia gerar comandos SQL em formato de texto, dessa forma, os comandos poderiam ser facilmente executados de forma manual no banco de dados desejado; ou o robô poderia conectar-se diretamente com o banco de dados utilizando uma biblioteca adicional do seu desenvolvimento.

2.2.2 Frameworks

A utilização de uma linguagem de programação não específica para criação de robôs possui alguns problemas, como por exemplo: a necessidade de incluir

diversas bibliotecas para as mais variadas tarefas, como acesso a página, parsing, extração e armazenamento de informações. Além disso, robôs são conhecidos por ser muito frágeis a mudanças na estrutura de HTML dos recursos acessados. Em linguagens compiladas como Java, qualquer mudança na implementação do robô força a uma nova compilação e até mesmo a reimplantação da aplicação (GLEZ-PENÑA et. al., 2014, tradução nossa).

Dessa forma, surgiram alguns frameworks que visam simplificar o processo de data scraping, entre elas estão Scrapy, Web-Harvest e jARVEST. Alguns dos *frameworks* disponíveis implementam a sua própria *Domain Specific Language* (DSL), que seria uma linguagem de programação desenvolvida especificamente para a atividade proposta pelo framework, esta que simplifica e diminui a quantidade de código que deve ser produzido para a implementação do robô.

No anexo A pode ser observado, por exemplo, a implementação de um robô construído com jARVEST utilizando a sua própria DSL, robô que irá acessar as duas primeiras páginas do site Kickstarter e buscar informações dos projetos recentemente lançados, código que foi desenvolvido por GITHUB.COM/MICHADA (2018).

2.2.3 Aplicações

A necessidade de criar um robô para a extração de informações de páginas nem sempre virá a ser suprida por um programador, para estes casos, existem as aplicações baseadas em desktop. Aplicativos para *Web Scraping* geralmente possuem diversas funcionalidades já embutidas na sua interface, como por exemplo: Navegador embutido, geração em formato texto como CSV, armazenamento em banco de dados, etc.

Entre algumas das aplicações com esse objetivo, no mercado podem ser encontrados exemplos como dexi.io, ParseHub, Scraper, etc. A aplicação dexi.io por exemplo, conforme o seu site ([2018]), possui interface web com diversas funcionalidades que facilitam no desenvolvimento do robô, onde: é possível verificar em tempo real o resultado de uma extração dentro da própria interface de construção de robôs; realizar conexão com a API documentada do aplicativo através de software

de terceiros para as mais diversas funcionalidades; usar integrações já prontas com outros aplicativos como Amazon S3 e Google Drive; entre outros.

Porém, a maioria das aplicações são comerciais e de código fechado, dessa forma, existem diversas limitações, como por exemplo, impossibilidade de alteração no código fonte, custo do licenciamento e acesso limitado as APIs disponibilizadas (GLEZ-PEÑA et. al., 2014, tradução nossa).

3 INTELIGÊNCIA ARTIFICIAL APLICADA AO RECONHECIMENTO DE IMAGENS

A inteligência artificial é amplamente conhecida como a área inserida no campo da ciência da computação que abrange a automação do comportamento inteligente (LUGER, 2013). Dessa forma, pode se entender que o reconhecimento de imagens, por ser tratar de um comportamento inteligente, também pode ser automatizado.

A Visão Computacional é uma subárea que se deriva da Inteligência Artificial, ela tem o objetivo de prover ao computador a habilidade de descrever o mundo que é visto em uma ou mais imagens e reconstruir as suas propriedades, como formas, iluminação, e distribuição de cores (SZELISKI, 2011, tradução nossa).

Para possível ao chegar ao nível de implementação existente atualmente na área de Visão computacional, se foi necessário a melhoria significativa dos recursos e técnicas disponíveis, como por exemplo, a ascensão das redes neurais e a grande quantidade de imagens rotuladas disponíveis na internet (SZELISKI, 2011, tradução nossa). Hoje a Visão computacional possui as mais variadas aplicações no mercado, estas que vão desde de a classificação de imagens até ao auxílio a médicos para a procura de câncer de difícil detecção em pacientes.

Figura 4 - Visão computacional aplicada a carros autônomos



Fonte: Mobileye ([20--]).

Outro ramo da indústria que já usufrui das tecnologias de visão computacional são os carros autônomos, tecnologia que possui capacidade de mudar totalmente a forma em que a sociedade vive. Os carros autônomos possuem diversos sensores, entre eles uma câmara acoplada a um computador de bordo que utiliza de inteligência artificial com um algoritmo de visão computacional, conforme verificado na figura 4, o algoritmo possui capacidade de detectar pedestres, sinais de trânsito, tráfego de carros, limites da estrada, etc.

Além disso, a área de visão computacional possui diversas outras aplicações, entre elas (SZELISKI, 2011, tradução nossa):

- a) reconhecimento de caracteres: leitura de caracteres em placas, rótulos, textos, etc;
- b) inspeção de máquinas: com a utilização de visualização de peças é possível garantir a qualidade do produto;
- c) varejo: no reconhecimento de objetos para a fila de *checkout*;
- d) modelagem 3D automatizada de construções: utilizada por exemplo pelo Bing Maps;
- e) captura de movimento: através da utilização de câmeras em diversos ângulos com o auxílio de marcadores nos atores para a detecção de movimento, de forma que possa ser utilizado posteriormente em por exemplo, animações 3D;
- f) vigilância: realizando monitoramento de intrusos, tráfego de veículos e piscinas públicas para encontrar vítimas de afogamento;
- g) reconhecimento de digitais e biometria;

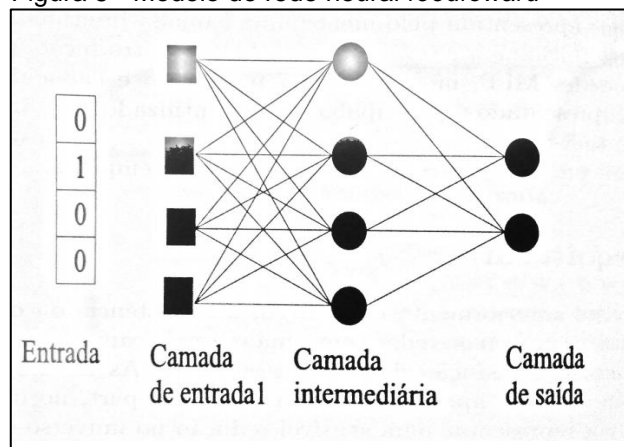
3.1 SISTEMAS DE CLASSIFICAÇÃO DE IMAGENS

Sistemas de classificação de imagens são compostos de funções matemáticas e alguns conceitos da computação, porém, quando aplicados corretamente, podem servir como base para um complexo sistema de visão computacional, como por exemplo, a categorização automática de fotos nas galerias dos celulares iPhone mais recentes. Dessa forma, podem identificadas algumas possíveis maneiras de se realizar classificação de imagens, como através das Redes Neurais Artificiais.

3.1.1 Redes neurais artificiais

A possibilidade de implementação de um raciocínio similar ao do ser humano em máquinas, assim criando mentes artificiais, é uma das apostas da tecnologia moderna para resolução dos mais diversos problemas da sociedade contemporânea. O conceito de mente artificial existe a mais de 50 anos, porém, vem se destacando recentemente no meio científico e tecnológico com base nas suas evoluções e aplicações a partir dos anos 90. Dessa maneira, ao longo dos anos, baseando se no sistema nervoso humano, surgiu o conceito de Redes Neurais Artificiais (SILVA; SPATTI; FLAUZINO, 2010).

Figura 5 - Modelo de rede neural *feedforward*



Fonte: Braga, Carvalho e Ludemir (2000).

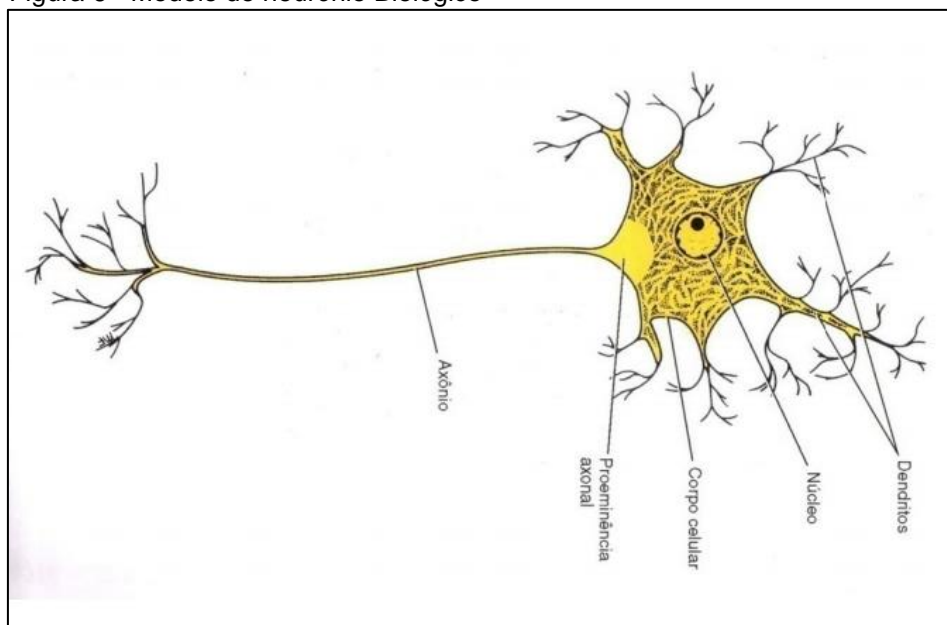
Conforme a figura 5, as redes neurais podem ser divididas em três tipos de camadas: Camada de entrada, é a camada responsável por receber os dados

externos, sendo que estes dados são geralmente normalizados; Camadas escondidas, são camadas que possuem os neurônios responsáveis por produzir o resultado extraído com base no objetivo da rede neural; Camada de saída, camada que também possui neurônios, responsável por produzir e exibir os resultados finais com origem nos processamentos realizados nos neurônios das camadas anteriores (SPATTI; FLAUZINO, 2010).

3.1.1.1 Neurônio Artificial

As Redes Neurais Artificiais são modelos computacionais que possuem similaridades com o que pode ser entendido hoje nos estudos de anatomia como sistema nervoso do ser humano. Dessa forma, assim como no sistema nervoso humano, a rede neural toma como sua unidade mínima e base para as suas operações o neurônio, conforme demonstrado na figura 6 (SILVA; SPATTI; FLAUZINO, 2010).

Figura 6 - Modelo de neurônio Biológico



Fonte: Snell (2011).

O sistema nervoso humano é um sistema embaraçado e organizado com milhões de neurônios, estes que são responsáveis pela recepção e transmissão dos estímulos externos, possibilitando ao corpo responder de forma adequada ao estímulo recebido (ANDRADE FILHO; PEREIRA, 2015).

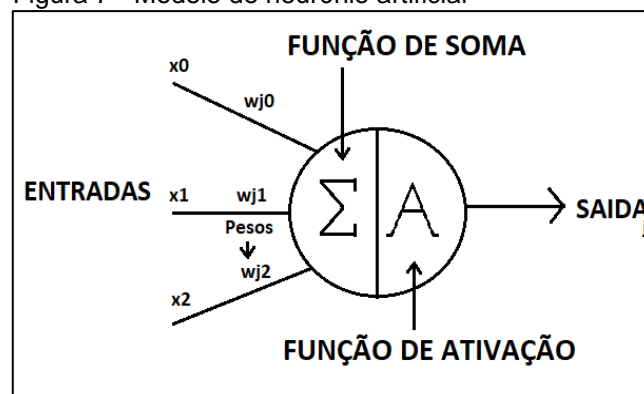
Inspirado pelo neurônio biológico, um neurônio artificial é composto por (LUGER, 2013):

- a) sinais de entrada: variável utilizada para cálculo do estado;
- b) conjunto de pesos: estes pesos descrevem a força da conexão do neurônio para com outros neurônios que este possui relacionamento;
- c) ativação ou estado: valor produzido pelo cálculo da função de ativação;
- d) função de ativação: função matemática responsável por produzir o resultado que indica a ativação do neurônio.

Cada neurônio deverá possuir uma função de ativação, esta que irá definir o estado do neurônio, chamado de ativação. Para gerar a ativação através do cálculo obtido pela função, o neurônio deverá receber um ou mais valores de entrada, este que poderá ter origem pela ativação de outros neurônios da rede, de camadas anteriores a qual este possui relacionamento, ou de origem externa através entrada da rede neural. (PALMA NETO; NICOLETTI, 2005).

A ativação do neurônio é calculada utilizando a soma dos valores de entrada ponderados, após o cálculo da ativação, o valor será enviado como entrada para os neurônios da próxima camada ou apresentada como saída da rede neural caso o neurônio pertencer a camada de saída (SILVA; SPATTI; FLAUZINO, 2010).

Figura 7 - Modelo do neurônio artificial



Fonte: Adaptado de Haikin (2009).

Além dos elementos citados na figura 7, o neurônio também possui implementações na qual utilizasse o *Bias*, este que pode ser definido como variável que define a inclinação para a ativação do neurônio. Assim como os pesos de cada conexão, o *Bias* pode ter o valor diferente para cada neurônio e também receber

modificações no seu valor durante o treinamento da rede (LUDWIG JUNIOR; COSTA, 2007).

Dessa forma, pode-se identificar a função (1) adaptada do livro de Flauzino (2010) para definição da saída de um neurônio simples. Na fórmula, pode ser identificado que o u seria a diferença do *Bias* para a soma das n entradas (x) com seus respectivos pesos (w) aplicados, após, o resultado é enviado para a função de ativação g que será responsável por calcular a saída final y .

$$u = \sum_{i=1}^n w_i x_i - \theta \quad (1)$$

$$y = g(u)$$

O objetivo de uma rede neural irá definir a quantidade de camadas e entradas que ela irá possuir. Uma rede neural que irá realizar a análise de uma imagem por exemplo, a quantidade de entradas possivelmente será a quantidade de pixels da imagem, dessa forma, a rede neural irá avaliar os valores dos pixels e enviar para as suas camadas para o processamento.

3.1.1.2 Arquiteturas de redes neurais

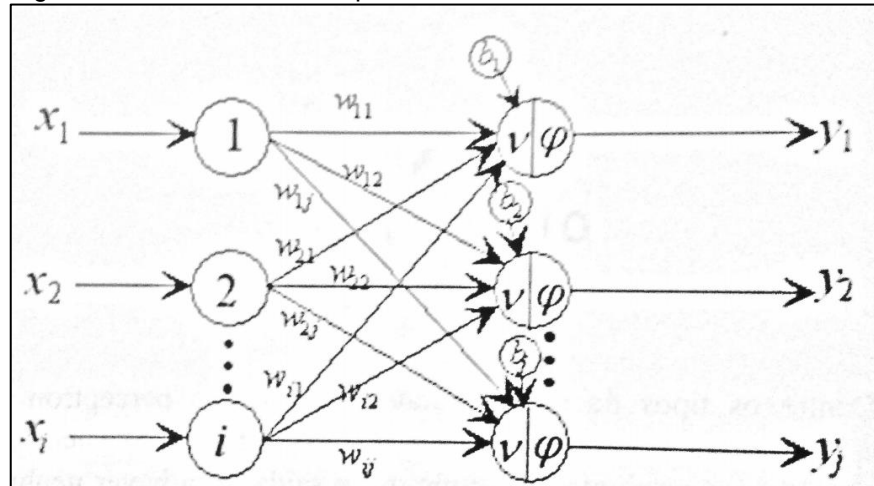
Existem algumas arquiteturas diferentes para a organização da rede neural, estas que variam a disposição dos neurônios e também a maneira que seus neurônios são conectados. Algumas das arquiteturas são: *feedforward* de camada simples (alimentação para frente), *feedforward* de camadas múltiplas, redes recorrentes e redes reticuladas (SPATTI; FLAUZINO, 2010).

3.1.1.2.1 Arquitetura *feedforward* de camada simples

Nas arquiteturas *feedforward* o fluxo de informações irá sempre na mesma direção, ou seja, iniciará na camada de entrada e terminará na camada de saída. Dessa forma, a arquitetura de camada simples possui apenas a camada de entrada e neurônios na camada de saída (SPATTI; FLAUZINO, 2010). Um exemplo dessa

arquitetura seriam as redes Perceptron, nelas todos os elementos da camada de entrada estão conectados a cada neurônio da camada de saída.

Figura 8 - Rede neural Perceptron



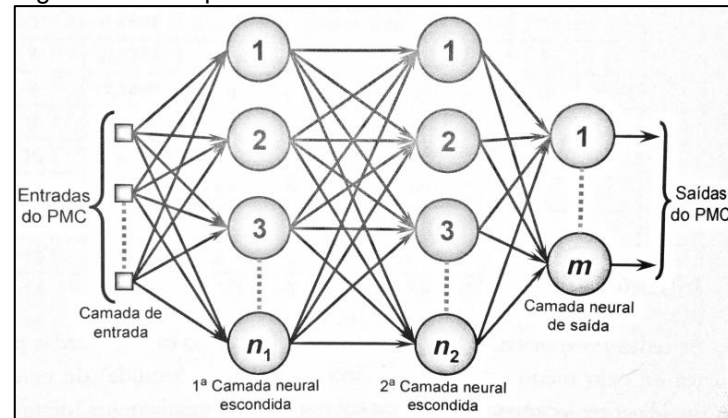
Fonte: Ludwig Junior e Costa (2007).

Conforme verificado na figura 8, a rede Perceptron possui a mesma quantidade de neurônios em ambas camadas, essa simplicidade acaba limitando a complexidade dos problemas para a sua aplicação, permitindo apenas ser aplicada em estruturas de decisões simples como classificação de padrões e filtragem linear (LUDWIG JUNIOR; COSTA, 2007).

3.1.1.2.2 Arquitetura feedforward de camadas múltiplas

Diferente da arquitetura de camada simples, a rede *feedforward* de camadas múltiplas possui uma ou mais camadas de neurônios escondidos, esta diferença possibilita que ela seja aplicada em problemas mais complexos, como classificação de padrões, identificação de sistemas, otimização, robótica, controle de processos, etc (SPATTI; FLAUZINO, 2010).

Figura 9 - Perceptron Multicamadas



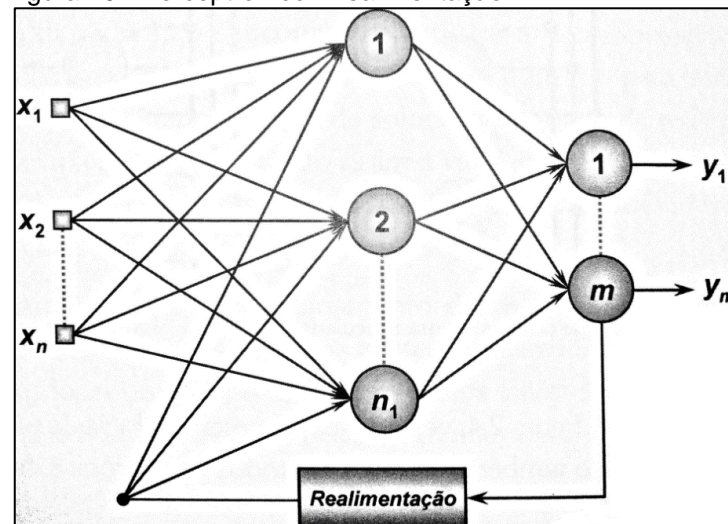
Fonte: Spatti e Flauzino (2010).

Conforme pode ser observado na figura 9, as redes Perceptron possuem uma implementação com esta arquitetura chamada de Perceptron Multicamadas (PMC), do inglês *Multilayer Perceptron* (MLP).

3.1.1.2.3 Arquitetura recorrente

As redes recorrentes são classificadas como redes para processamento dinâmico, isso ocorre pois em sua arquitetura as saídas da rede são utilizadas para realimentação de outros neurônios da mesma rede, criando assim a variação em sua resposta. Este tipo de rede pode ser utilizado em alguns tipos específicos de aplicações como previsão de tempo, controle de processos, entre outros (SPATTI; FLAUZINO, 2010).

Figura 10 - Perceptron com realimentação



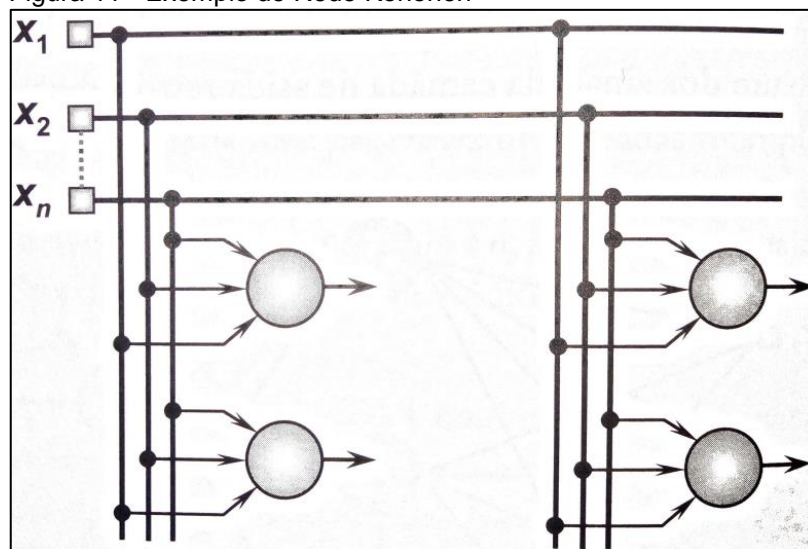
Fonte: Spatti e Flauzino (2010).

Conforme pode ser observado na figura 10, existe também uma implementação de redes Perceptron com a arquitetura de realimentação, esta que se realimenta a partir da camada de saída.

3.1.1.2.4 Arquitetura com estrutura reticulada

Redes reticuladas possuem uma arquitetura diversificada na localização dos neurônios, isso ocorre com o propósito de extração de informações em camadas da rede, informações essas que são utilizadas para o seu posterior treinamento por método competitivo. Esse tipo de rede possui uma gama de aplicações diferenciadas, desde a problemas de agrupamento até reconhecimento de padrões, grafos, entre outros (SPATTI; FLAUZINO, 2010).

Figura 11 - Exemplo de Rede Kohonen



Fonte: Spatti e Flauzino (2010).

Conforme verificado na figura 11 na rede *Kohonen*, principal rede da arquitetura, a estrutura diversificada permite que neurônios de saídas possam ser encontrados em camadas intermediárias, por exemplo.

3.1.1.3 Redes Neurais Convolucionais

Redes neurais convolucionais, do inglês *convolutional neural networks* (CNN), são redes neurais especializadas em processamento de dados em grade, ou

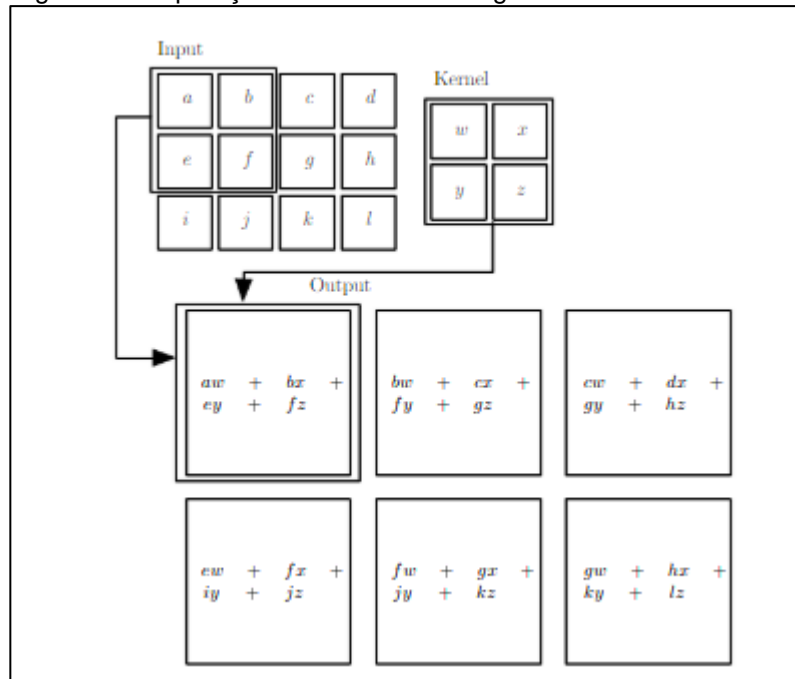
matriz, sejam essas grades de uma ou mais dimensões. Diante disso, esse tipo de rede neural tem sido aplicado com sucesso em processamento de imagens. (GOODFELLOW; BENGIO; COURVILLE, 2016, tradução nossa). Além disso, esse tipo de rede possui uma alta tolerância a variações no posicionamento e tamanho, além de outros tipos de distorção de imagens (HAYKIN, 2009, tradução nossa).

Dessa forma, conforme Haykin (2009), para obter a classificação de uma imagem ou outro tipo de dados em grade, a rede neural convolucional realiza a maior parte das suas funções através de 3 princípios:

- a) extração de características: cada neurônio obtém dados de entrada do neurônio da camada anterior onde se observa os valores periféricos a ele, realizando assim a extração de características locais, dessa forma, a localização da característica deixa de ser importante, desde que a posição relativa as demais características extraídas continue a mesma.
- b) mapeamento de características: cada camada convolucional da rede é composta de múltiplos mapas de características.
- c) subamostragem ou *subsampling*: cada camada convolucional é seguida de uma camada que realiza a média dos neurônios locais para a construção de uma grade menor, esta que pode ser utilizada posteriormente em uma nova camada convolucional. Essa operação aumenta a tolerância a distorções.

A extração de características é realizada com a operação de convolução, obtida através da utilização de um filtro, também conhecido como *kernel* ou detector de características, este que irá avaliar os valores da grade de entrada e gerar um mapa da característica sendo avaliada, conforme pode ser observado na etapa de convolução da imagem 12 (GOODFELLOW; BENGIO; COURVILLE, 2016, tradução nossa).

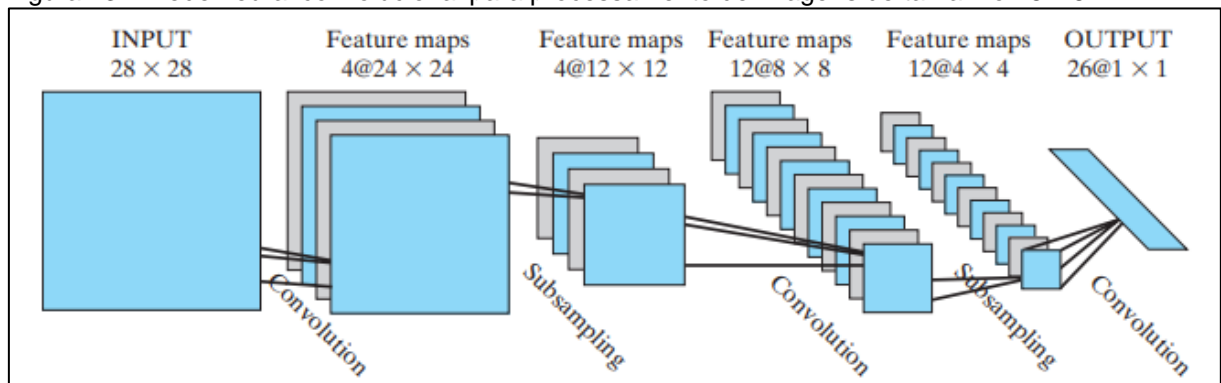
Figura 12 – Operação convolucional em grade de uma dimensão



Fonte: Goodfellow, Bengio e Courville (2016).

Na figura 12, por exemplo, podemos observar que foi obtida um mapa de características 3x2 após a aplicação da convolução com um *kernel* 2x2 em uma grade de valores 4x3, porém, algumas outras maneiras de realizar a operação pode resultar em mapas de características diferentes, por exemplo, ao utilizar um *kernel* 3x3 é possível obter uma grade de saída 4x3.

Figura 13 – Rede neural convolucional para processamento de imagens de tamanho 28x28



Fonte: Haikin (2009).

Dessa forma, conforme pode ser observado na imagem 13, uma rede neural convolucional pode possuir diversas camadas de convolução, onde o mapa de características da camada anterior servirá de entrada para camada de convolução

seguinte, após passar pela camada de subamostragem (HAYKIN, 2009, tradução nossa).

3.1.1.4 Treinamento de redes neurais

Uma das grandes capacidades das redes neurais que as tornam tão atrativas é a possibilidade de aprendizado baseando-se em um conjunto de amostras apresentadas. Dessa forma, após o treinamento com uma quantidade significativa de amostras, a rede neural é capaz de produzir uma saída que condiz com os exemplos apresentados (SPATTI; FLAUZINO, 2010).

O aprendizado de uma rede neural pode ser obtido basicamente através de dois processos principais distintos, aprendizado supervisionado e aprendizado não supervisionado (LUDWIG JUNIOR; COSTA, 2007).

No método supervisionado de treinamento é disponibilizado um conjunto de amostras com as suas devidas saídas, dessa forma, as entradas são processadas pela rede neural e após será aplicado um algoritmo para avaliar a diferença da saída recebida pela rede com a saída do conjunto de amostra. Desse modo, a rede neural irá posteriormente utilizar um algoritmo específico para cada tipo de rede chamado de Algoritmo de Aprendizado, este que irá realizar os ajustes necessários nas variáveis da rede (LUDWIG JUNIOR; COSTA, 2007). Os pesos das conexões e *Bias* serão continuamente ajustados pelo Algoritmo de Aprendizagem até que a defasagem entre saída da rede e da amostra estejam em valores aceitáveis, isso será definido com base no objetivo definido para a rede em questão (SPATTI; FLAUZINO, 2010).

Em contrapartida, o método não supervisionado possui também um conjunto de amostras, porém, não são disponibilizadas as saídas correspondentes de cada amostra. Dessa forma, redes que utilizam desse método de aprendizagem são utilizadas para classificação de elementos, a rede adapta seus neurônios para realizar a classificação dos valores utilizando apenas as entradas recebidas, e por meio de processos de competição e cooperação, consegue obter os resultados desejados (LUDWIG JUNIOR; COSTA, 2007).

Normalmente as amostras disponibilizadas são previamente divididas em dois subconjuntos, estes são geralmente chamados de subconjunto de treinamento e subconjunto de validação ou teste. O primeiro, subconjunto de treinamento, possui o

exclusivo objetivo de realizar o treinamento da rede neural em junto ao Algoritmo de Aprendizagem, então, os elementos que o compõe serão selecionados aleatoriamente do conjunto total. Dessa forma, as amostras restantes, cerca de 10% a 40%, serão utilizadas para validar o aprendizado da máquina, este subconjunto não é utilizado na etapa de treinamento, com isso, é possível verificar a capacidade de generalização junto a valores de entrada desconhecidos (SPATTI; FLAUZINO, 2010).

Além dos dois métodos de treinamento principais, existem algumas variações do método supervisionado que também podem ser aplicadas, como é o caso do treinamento com reforço, a qual constantemente avalia a defasagem entre saída obtida e esperada. Outros dois métodos derivados conhecidos são aprendizagem usando lote de padrões e aprendizagem usando lote padrão-por-padrão, o primeiro leva em consideração todas as amostras antes de realizar a alteração dos pesos de conexão e *bias*, o segundo, diferente do primeiro, realiza a avaliação em cada amostra apresentada, porém, pode realizar o descarte das alterações (SPATTI; FLAUZINO, 2010).

4 RECONHECIMENTO ÓTICO DE CARACTERES

Nos dias atuais, com o advento da tecnologia, muitos dos meios de comunicação tradicionais tiveram que se adaptar à nova realidade e tendências. Os jornais impressos por exemplo, segundo a *World Association of News Papers* (WAN), existem a mais de 400 anos, porém, a criação da internet e dos computadores fez o jornal ser distribuído primeiramente de forma digital.

Porém, alguns meios de informação são mais difíceis de se adaptar à nova realidade, a literatura antiga e contemporânea, por exemplo, necessita que todos os livros, artigos, publicações em papel sejam manualmente digitalizadas. Dessa forma, ao longo dos anos, o Reconhecimento Ótico de Caracteres, surgiu para auxiliar os envolvidos na interpretação automática de todos os meios impressos.

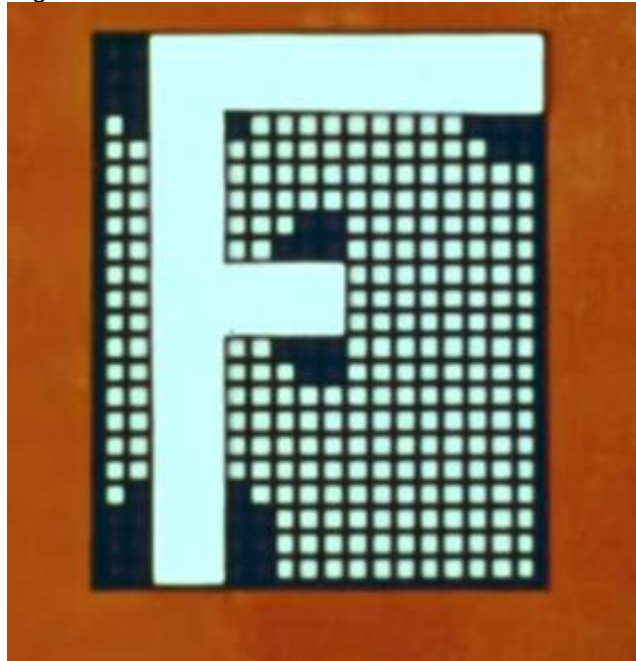
O surgimento de tecnologias OCR pode ser identificado a muito tempo atrás, como por exemplo, através da máquina criada pelo Dr. Edmund Fournier d'Albe, esta que analisava as variações das ondas da luz quando atravessavam uma folha de papel em contato com o elemento químico Selênio, dessa forma, era possível emitir um som diferente para cada letra lida pela máquina, que quando estudado corretamente, tornava possível a leitura de um texto por um deficiente visual por exemplo (DALBE, 1914).

A evolução da tecnologia permitiu avanços da forma como o reconhecimento dos caracteres era realizado. Aproximadamente nos anos 60 a 70 as OCR começaram a ser utilizadas pelos sistemas de correspondências para aumentar a velocidade da organização das correspondências. Conforme indicado pelo Serviço de Postagem dos Estados Unidos, as OCR como a do Serviço de postagem poderiam ler em média 42 mil endereços por hora, esse sistema utilizava um scanner que lia os pontos pretos e brancos da área de endereço de uma carta, tentando encontrar as formas geométricas que cada letra possuía para identifica-la e assim formar todo o texto dos endereços, conforme pode ser observado na imagem 14 (READING...,1970).

Porém, as tecnologias OCR ainda possuíam um longo caminho pela frente até se tornarem o que são hoje, isso porque os documentos impressos utilizados até então eram escritos com fontes padronizadas produzidas por máquinas de escrever

ou computadores, não sendo possível por exemplo, identificar caracteres escritos na mão.

Figura 14 - Reconhecimento de formas das letras



Fonte: Serviço postal dos EUA (1970).

A evolução da inteligência artificial nos últimos anos permitiu uma evolução significativa na quantidade de aplicações para uma OCR, isso porque atualmente a maioria das OCR utilizam de *machine learning* e *data mining*. Dessa forma, a evolução das OCR permitiu aplicações em necessidades mais complexas como: produção de fala sintetizada para cegos; reconhecimento de placa de carros por radares eletrônicos ou até mesmo tradução de placas em tempo real para turistas através do Google Tradutor, conforme demonstrado na figura 15.

Figura 15 - Google Tradutor, tradução em tempo real



Fonte: Verghese (2015).

Dessa forma, diversas plataformas comerciais de visão computacional surgiram nos últimos anos para suprir as novas necessidades de inteligência artificial para reconhecimento de imagens, entre elas, IBM Watson da IBM, Windows Azure da Microsoft, Amazon Rekognition da Amazon e Google Cloud Vision da Google. As plataformas, além possuírem algumas funcionalidades distintas para o reconhecimento da imagem, são capazes de realizar o OCR dos caracteres das imagens disponibilizadas.

4.1 IBM WATSON

IBM Watson é uma plataforma comercial em nuvem da IBM, ela visa disponibilizar produtos baseados em inovações de *machine learning* para seus mais variados clientes aplicarem em suas diferentes necessidades. Por ser uma plataforma em nuvem, a integração com os produtos dos clientes são realizadas através de *web services*, permitindo a escalabilidade dos recursos conforme o necessário, pois a plataforma é dividida em vários produtos, dessa forma eles podem ser adquiridos em demanda. Watson Assistance por exemplo, é um produto disponível, ele possui ferramentas para criação de robôs de conversas, ou *chatbots*, estes que podem ser utilizados para a automatização de atendimento ao cliente. Outro exemplo é o serviço 'Text to Speech', este que visa disponibilizar a produção de voz sintetizada em produtos a partir de textos, possibilitando a melhor interação do cliente com o produto, ou até mesmo, melhorar a acessibilidade da interface de usuário para que pessoas com deficiência visual possam utilizar (IBM, [2018]).

4.1.1 Reconhecimento de texto com IBM Watson

O produto para visão computacional do IBM Watson chama-se Visual Recognition, ele possui modelos de redes neurais treinados para realizar a extração de cenas, objetos, faces, cores, comidas e outras possíveis características de uma imagem. Além disso, é possível disponibilizar amostras para um treinamento personalizado, de forma que o cliente da IBM possa obter os resultados para um campo de conhecimento específico. Dessa forma, o reconhecimento de texto de imagens está disponível através do modelo de detecção pré treinado chamado *Text Model*, do português, Modelo de Texto.

Além disso, é possível utilizar a plataforma de visão computacional da IBM para gerar modelos de Core ML, que permite a execução dos modelos de redes neurais diretamente em celulares com sistema operacional IOS (IBM, [2018]).

4.2 WINDOWS AZURE

A Windows Azure é uma plataforma em nuvem da Microsoft de classificação empresarial que possui soluções para as mais variadas necessidades de uma empresa, desde a hospedagem de aplicações web e ferramentas para desenvolvimento de sistemas em nuvem até a serviços avançados de inteligência artificial baseados em redes neurais.

4.2.1 Reconhecimento de texto com Windows Azure

Segundo a Microsoft ([2018]), *Computer Vision* é um dos serviços disponibilizados na plataforma Azure para a realização de Análise de imagens, podendo assim a partir da análise extrair informações como descrições de objetos da cena, detecção de *tags*, identificar se possui conteúdo adulto, realizar o reconhecimento facial, entre outras coisas. Outro grande aspecto do serviço é o OCR, onde além de extrair das imagens os caracteres impressos, também é possível identificar escrita à mão.

4.3 AMAZON REKOGNITION

Amazon Rekognition é um serviço em nuvem da Amazon, este disponibiliza API de análise de imagens e vídeos para serem utilizados em aplicativos de terceiros. Através do serviço, é possível analisar objetos, pessoas, texto, cenas e atividades, ou também verificar se possui conteúdo não adequado. Além das funcionalidades citadas, é possível realizar também o reconhecimento e comparação facial (AMAZON, [2018]).

4.3.1 Reconhecimento de texto com Amazon Rekognition

O recurso de API *DetectText* permite realizar a extração do texto de imagens em png e jpg, bastando enviar a imagem através da API para receber uma lista com as palavras detectadas e suas especificações, como confiança, texto detectado, posição na imagem, etc.

4.4 GOOGLE CLOUD VISION

O Google Cloud Vision é uma API da Google que possui modelos treinados ou permite realizar o treinamento de modelos para realizar análise de imagens. Através da API REST é possível categorizar as imagens, detectar objetos e rostos e também extrair os textos impressos das imagens.

Para realização de treinamento de novos modelos de qualidade, é disponibilizado o recurso *'AutoML Vision Beta'*, este que pode ser utilizado até por desenvolvedores com pouco conhecimento em treinamento de redes neurais. Dessa forma, para o treinamento, basta realizar o envio das imagens e aplicar os marcadores desejados, após, o *'AutoML Vision'* treinará um novo modelo que poderá ser escalonado conforme a necessidade para se adaptar a demanda da interface.

4.4.1 Reconhecimento de texto com Google Cloud Vision

A API da plataforma possui funcionalidade de extração de textos escritos a mão e impressos, assim como também a possibilidade de extrair das imagens características adicionais como logotipos, moderação de conteúdo, cores dominantes, sugestões de recorte entre outras coisas (GOOGLE, [2018]).

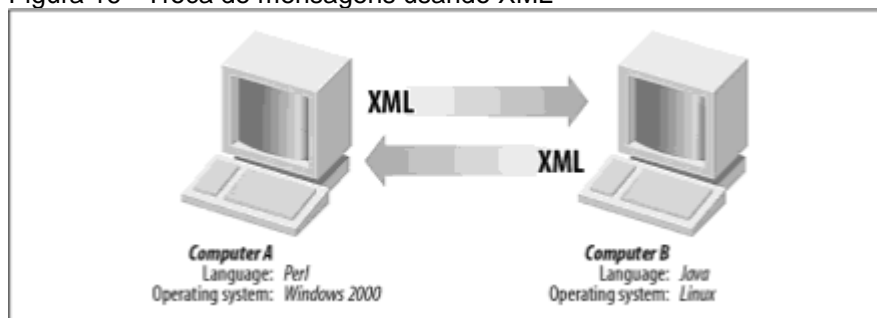
5 WEB API

Softwares armazenados em nuvem, como plataformas de visão computacional, possibilitam o acesso as suas funcionalidades através da utilização de Web API.

Os softwares são geralmente utilizados por usuários comuns, porém, nada impede que estes também possam ser de utilidade para outras aplicações, dessa forma, é possível evitar o retrabalho através de utilização de funções presentes nos mais diversos *softwares* disponíveis no mercado. Para que seja possível a utilização de programas de terceiros, surgiram as API, estas que são interfaces que permitem a comunicação e programação destes *softwares* de terceiros (BIEHL, 2016, tradução nossa).

Existem algumas formas de implementar uma API, porém, através do surgimento da Internet, a Web API ganhou força através dos chamados *web services*, este que pode ser definido como qualquer serviço disponibilizado na internet que utiliza troca de mensagens padronizada, geralmente através do XML, para realizar a comunicação sem a presença de uma linguagem ou sistema operacional (CERAMI, 2002, tradução nossa).

Figura 16 - Troca de mensagens usando XML



Fonte: CERAMI, Ethan (2002).

Como pode ser verificado na figura 16, os *web services* permitem que uma máquina utilizando por exemplo, o sistema operacional Windows 2000 executando um programa escrito em linguagem Perl possa se comunicar com uma segunda máquina que utilize a linguagem Java e sistema operacional Linux, usando apenas o XML em uma estrutura padronizada para a linguagem da comunicação com intermédio da rede de computadores. Dessa forma, a flexibilidade de *web services* permitem também a

diminuição de retrabalho em relação as demais possíveis formas de se implementar uma API.

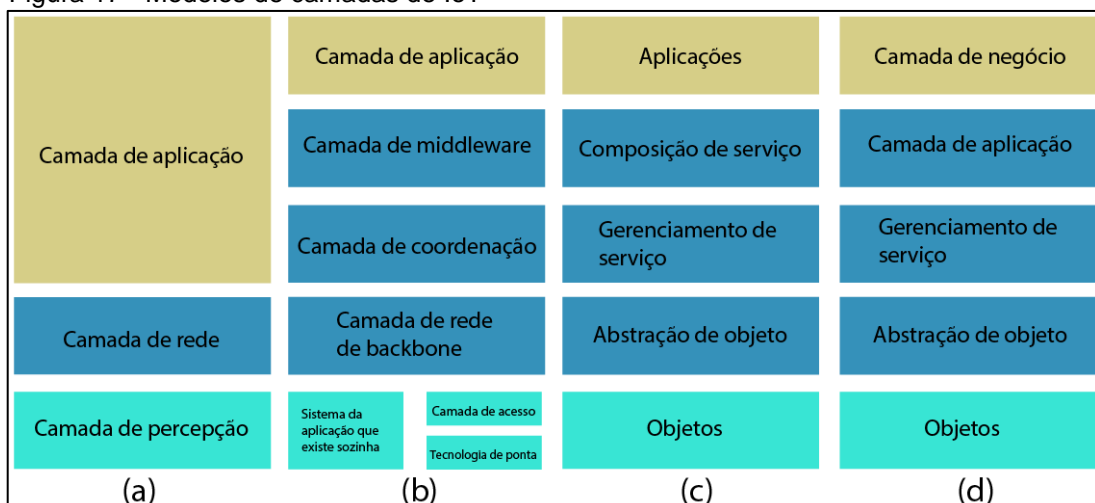
Web services podem ser implementados através de 2 protocolos diferentes, *Simple Object Access Protocol* (SOAP) e *Representational State Transfer* (REST). Tipicamente, SOAP é um dialeto XML com um documento especificando a estrutura necessária para que uma requisição seja definida como uma comunicação SOAP, dessa forma, o cliente enviará uma requisição HTTP para o servidor utilizando uma estrutura XML apropriada e também receberá uma resposta com a mesma estrutura. Por outro lado, REST não possui um padrão estritamente definido no protocolo para que possa ser facilmente explicado, porém, ao contrário de SOAP, REST geralmente utiliza o formato JSON para a estruturação da mensagem trocada entre cliente e servidor. Outra grande diferença do REST, é que ele não apenas utiliza o HTTP como meio para envio da mensagem, mas também para especificação de padrões da sua comunicação (KALIN, 2013).

6 INTERNET DAS COISAS

A sociedade contemporânea está se tornando dependente de dispositivos eletrônicos, conforme a *International Telecommunication Union* (2005) estima-se que já em 2005 já existiam cerca de 2 bilhões de *smartphones* em todo mundo, número que passou para mais de 5 bilhões em 2017 segundo a GSMA (2017). Dessa forma, ao longo dos anos, novas formas de comunicação eletrônica surgiram, não apenas entre pessoas, mais também entre pessoas e dispositivos ou até mesmo entre os dispositivos.

Dessa forma, surgiu o que é conhecido hoje como Internet das coisas, do inglês Internet of Things (IoT), o conceito busca não apenas a comunicação entre computadores, celulares e tablets, mas todas as “coisas” da rotina da sociedade moderna. A Internet das Coisas prega a integração tecnológica para que qualquer pessoa com um *smartphone* possa por exemplo, verificar detalhes de uma propaganda que está passando em um *display* sem precisar acessar um *site* específico, mas que isso seja realizado através da comunicação entre o *display* da propaganda e o *smartphone* (MCEWEN; CASSIMALLY, 2014, tradução nossa). Nesse sentido, a Internet das coisas é uma evolução tecnológica que representa o futuro da computação eletrônica, porém, o seu desenvolvimento depende da inovação tecnológica nos mais diversos dispositivos eletrônicos, desde de sensores sem fio a produtos com nanotecnologia (INTERNATIONAL TELECOMMUNICATION UNION, 2005, tradução nossa).

Figura 17 - Modelos de camadas de IoT



Fonte: Adaptado de Al-Fuqaha et. al. (2015).

A IoT deve ser capaz de conectar trilhões de dispositivos utilizando a internet, dessa forma, é necessário padronizar o conceito em algumas camadas. Atualmente não existe um modelo internacional para internet das coisas, porém pode ser realizado alguns relacionamentos entre os modelos existentes, conforme pode ser observado na imagem 17, onde (a) seria o modelo de 3 camadas; (b) modelo baseado em *middleware*; (c) modelo baseado na Arquitetura Orientada a Serviço, do inglês Service Oriented Architecture (SOA) e (d) modelo de 5 camadas (AL-FUQAHA et. al., 2015, tradução nossa).

Dessa forma, analisando os modelos de 5 camadas, podem ser obtidas as seguintes.

6.1 CAMADA DE OBJETO

A Camada de Objeto também pode ser chamada de Camada de Percepção, possui dispositivos físicos que irão coletar e processar dados. Esta camada inclui sensores e atuadores para as mais variadas tarefas, como detectar vibração, temperatura, localização, etc. A camada possui a funcionalidade de digitalizar e enviar os dados para a Camada de Abstração (AL-FUQAHA et. al., 2015, tradução nossa).

6.2 CAMADA DE ABSTRAÇÃO

Esta camada pode realizar algumas funções de computação em nuvem e tratamento de informações, porém, a principal funcionalidade da Camada de Abstração é realizar o envio das informações obtidas na Camada de Objeto para a Camada de Gerenciamento de Serviço, isso é obtido através de algumas tecnologias como RFID, 3G, GSM, Wifi, Bluetooth de baixa energia, etc (AL-FUQAHA et. al., 2015, tradução nossa).

6.3 CAMADA DE GERENCIAMENTO DE SERVIÇO

A Camada de *Middleware* ou Camada de Gerenciamento de Serviço realiza o ligamento entre um serviço e seu solicitador. Ela permite aos programadores

utilizarem uma interface heterogenia não se preocupando com o sistema operacional ou hardware dos dispositivos envolvidos (AL-FUQAHA et. al., 2015, tradução nossa).

6.4 CAMADA DE APLICAÇÃO

A Camada de Aplicação oferece para o cliente os recursos solicitados, por exemplo, ela será a responsável por entregar a temperatura na sala caso o cliente solicitar a informação. Camada essa que permite o desenvolvimento de carros, casas e prédios inteligentes (AL-FUQAHA et. al., 2015, tradução nossa).

6.5 CAMADA DE NEGÓCIO

A Camada de Negócio é responsável pela análise de todas as demais camadas, isso porque, além de realizar a avaliação dos dados obtidos em cada camada com os dados desejados, ela também produz gráficos, fluxos e relatórios que permitem a tomada de decisão a longo prazo. Além disso, a camada suporta a manutenção das demais camadas, provendo controles aos administradores do sistema para que as decisões tomadas possam ser aplicadas (AL-FUQAHA et. al., 2015, tradução nossa).

7 TRABALHOS CORRELATOS

Devido ao crescimento da procura por aplicações que facilitam o dia a dia através de conceitos de OCR e Internet das Coisas, se pode verificar diversos trabalhos em volta dos temas.

7.1 PROTÓTIPO DE APLICATIVO ANDROID PARA EXTRAÇÃO DE TEXTO EM IMAGENS PARA BUSCA SEMÂNTICA SOBRE RÓTULOS DE CERVEJAS

Projeto desenvolvido por Luana Gomes Silva em 2014 para adquirir o grau de bacharelado em Ciência da Computação na Universidade do Extremo Sul Catarinense.

Conforme descrito por Silva (2014), a proposta do trabalho seria a utilização do motor de OCR *open source* da Google chamado Tesseract para realizar o treinamento de uma rede neural OCR para a leitura de rótulos de cervejas através de imagens enviadas por usuário, após, seriam utilizados os caracteres obtidos para procurar informações sobre a cerveja no motor de busca do Google e mostrar as informações encontradas para o usuário através de um aplicativo para *smartphones*.

Dessa forma, o resultado do trabalho foi um aplicativo para Android que captura uma foto da cerveja, realiza a normalização da imagem para facilitar a detecção e envia a imagem para uma API REST em um servidor desenvolvido, este que irá realizar o OCR e após retornar as informações encontradas no motor de busca. Além disso, chegou na conclusão o processo de OCR do aplicativo era sensível a diversos fatores externos, como iluminação, alinhamento do texto e rotação da imagem ou da câmera, dessa forma, foi desenvolvido uma função auxiliar para o usuário informar o texto da cerveja caso a o motor OCR não o reconhecer automaticamente. Conforme descrito, em testes realizados com o rótulo de 10 cervejas, houve uma taxa de reconhecimento de 60%, e dentre as 6 cervejas com texto reconhecido, 5 delas trouxeram informações relevantes após a consulta no motor de busca.

Para trabalhos futuros, entre os itens sugeridos por Silva, estão o desenvolvimento de um motor de busca próprio, além também da aplicação de mais

técnicas de pré-processamento de imagens para melhorar a taxa reconhecimento, entre outros.

7.2 PROPOSTA DE ARQUITETURA DE UM SISTEMA COM BASE EM OCR NEURONAL PARA RESGATE E INDEXAÇÃO DE ESCRITAS PALEOGRÁFICAS DO SEC.XVI AO XIX

Dissertação de mestrado em Engenharia Elétrica produzido por Fábio Lúcio Lopes de Mendonça na Universidade de Brasília.

De acordo com Mendonça (2008), o trabalho visava implementar uma arquitetura de sistema para tratamento e realização da leitura OCR de documentos paleógrafos do século XVI ao XIX usando redes neurais artificiais.

Dessa forma, como resultado do estudo, foi desenvolvido um protótipo de *software* que permitiria ao usuário realizar a segmentação manual do texto para o treinamento da rede neural, assim, a rede treinada poderia ser utilizada posteriormente de dentro do sistema modular para a realização do OCR nos documentos digitalizados. Além disso, Mendonça conclui que a arquitetura proposta é funcional, porém, eram necessários estudos mais profundos sobre reconhecimento de escritas paleografadas do século XVI ao XIX, pois estes tipos de caracteres requerem técnicas mais complexas. Dessa forma, para trabalhos futuros, foram citadas necessidades de aprimorar as técnicas de extração, como o pré-processamento, entre outros.

7.3 RECONHECIMENTO DE DÍGITOS EM IMAGENS DE MEDIDORES DE CONSUMO DE GÁS NATURAL UTILIZANDO TÉCNICAS DE VISÃO COMPUTACIONAL

Dissertação de mestrado em Engenharia de Sistemas Computacionais produzido por Julio Cesar Gonçalves na Universidade Tecnológica Federal do Paraná.

Conforme descrito por Gonçalves (2016), o trabalho estuda técnicas para processamento de imagens e classificação de padrões para o reconhecimento de caracteres. Dessa forma, ele propõe o reconhecimento de dígitos dos contadores de consumo presentes nos medidores de gás natural utilizando tecnologias de visão computacional. Porém, diferente da maioria dos trabalhos apresentados na literatura,

Gonçalves utilizou um banco de imagens com cores para o treinamento do classificador.

Durante o trabalho, Gonçalves chegou à conclusão que, ao utilizar classificadores com redes neurais do tipo *Extreme Learning Machine* (ELM) no reconhecimento de dígitos segmentados, se obteve um desempenho superior sobre os outros dois tipos de algoritmos de classificadores testados no trabalho, *K-Nearest Neighbors* (KNN) e *Single Vector Machine* (SVM). Além disso, a metodologia utilizada mostrou-se promissora, já que obteve uma taxa de 95% de acerto no reconhecimento dos caracteres através da utilização de 903 imagens de medidores de gás para o treinamento da rede. Gonçalves conclui sugerindo alguns trabalhos futuros, entre eles estão a utilização de outros tipos de imagens no treinamento da rede, além do teste do modelo em dispositivos com *hardware* de baixo custo, como *smartphones*.

7.4 SISTEMA DE DIGITALIZAÇÃO, DETECÇÃO E CLASSIFICAÇÃO AUTOMÁTICA DE RECIBOS EM DISPOSITIVOS MÓVEIS

Dissertação de mestrado em Engenharia de Redes de Comunicação e Multimídia produzido por Miguel João Rodrigues Batista no Instituto Superior de Engenharia de Lisboa.

De acordo com Batista (2012), seria utilizada a tecnologia dos *smartphones* que estão disponíveis para a maioria da população de forma que o armazenamento de recibos possa ser simplificado. Dessa maneira, o trabalho propõe um aplicativo para *smartphone* que realiza o reconhecimento dos caracteres de recibos assim como também a sua classificação utilizando aprendizagem incremental. Dessa forma, seria possível gerir os recibos de forma automática e inteligente, isso seria realizado utilizando a câmera do *smartphone* junto a algoritmos de realidade aumentada. Além disso, Batista também propôs realizar a avaliação dos algoritmos utilizados durante o desenvolvimento do trabalho. Como resultado, foi obtida uma aplicação para dispositivo móveis que consegue processar uma imagem de recibo e normaliza-la, após, o texto do recibo é extraído através de OCR utilizando o motor open source Tesseract da Google, sequencialmente o recibo passa por um classificador que irá armazená-lo em uma categoria específica junto as informações extraídas.

8 IMPLEMENTAÇÃO DE PROTÓTIPO BASEADO NA TECNOLOGIA OCR APLICADA AO RECONHECIMENTO DE RÓTULOS PARA BUSCA EM BANCO DE DADOS

O trabalho realizado tem como objetivo um protótipo de aplicativo web que permita o cadastro de informações de produtos junto com uma foto do rótulo, desta forma, o rótulo poderia ser utilizado futuramente para auxiliar o acesso as informações cadastradas. A metodologia proposta em teoria pode ser aplicada a qualquer tipo de produto que possua rótulo com texto, porém, foi definido a utilização de cervejas para o protótipo deste projeto.

O protótipo deve armazenar diversas informações da cerveja desejada, dentre as informações estão cervejaria, ingredientes, estilo de cerveja, copos ideais para utilização, entre outros. Porém, o processo de consulta de informações já armazenadas no banco deve ser facilitado, isso é possível ao utilizar a câmera do dispositivo do usuário para enviar uma foto do produto que o usuário tem em mãos, dessa forma, uma imagem do rotulo é enviada para os servidores do aplicativo para o reconhecimento, caso desejado.

8.1 METODOLOGIA

Primeiramente, observou-se a necessidade de obter dados de produtos para armazenamento um banco de dados com dados de cervejas, dessa maneira, o protótipo final iria possuir uma quantidade considerável de informações. Assim sendo, foram estudadas maneiras de realizar extração de dados com base nos sites disponíveis na internet.

Em segundo lugar, com o intuito de adquirir os conhecimentos necessários do processo de OCR, foi realizado uma pesquisa bibliográfica focando em tecnologias de visão computacional e extração de caracteres de textos impressos, dessa forma, foi possível obter um bom entendimento de como funcionam tecnologias de OCR baseadas em redes neurais, além disso, foi possível obter conhecimento para o desenvolvimento de uma CNN utilizada posteriormente no projeto.

Em terceiro lugar, após um bom entendimento das tecnologias de visão computacional, foram pesquisadas algumas informações sobre plataformas

comerciais de visão computacional para a utilização no trabalho, visto que não será desenvolvido um mecanismo de OCR a partir do zero.

Em quarto lugar, foi desenvolvido uma pesquisa bibliográfica de alguns dos conceitos norteadores do projeto do protótipo. Dessa forma, uma pesquisa sobre WEB API foi realizada, pois, por exemplo, umas das funcionalidades do protótipo seria uma interface em nuvem para ele possa ser utilizado em qualquer dispositivo com acesso a internet. Além do mais, “Internet das Coisas” foi uma das grandes inspirações para o desenvolvimento do trabalho, dessa forma, foi também pesquisado sobre o assunto para que o projeto tivesse fortes bases computacionais.

Sequencialmente, foram verificados alguns trabalhos correlatos para se obter conhecimento necessário de algumas maneiras para alcançar a conclusão do protótipo, e além disso, as informações seriam utilizadas para comparações com o resultado final do projeto.

Dessa forma, para o desenvolvimento do protótipo, foram definidas e executadas as etapas abaixo:

- a) verificou-se maneiras alternativas ao OCR para reconhecimento de rótulos.
- b) foram realizados testes nas plataformas de visão computacional para ser verificada qual a com melhor precisão para realização de OCR em rótulos de produtos.
- c) foi desenvolvido um núcleo do protótipo, este que permite acesso as suas funcionalidades via API REST
- d) criada uma interface gráfica para a aplicação através do desenvolvimento de um aplicativo com tecnologias da WEB.
- e) foram obtidas informações cadastrais sobre cervejas para inclusão no banco de dados através dos métodos de extração estudados.
- f) foi realizada uma análise das limitações do protótipo obtido e possíveis melhorias.

8.1.1 Métodos alternativos ao OCR para detecção de rótulos

A utilização de OCR para a detecção de rótulo pode ser considerada uma versão simplificada da detecção do objeto baseado na extração de características,

dessa forma, uma possível melhoria para o método proposto pode ser implementada a partir do momento que o algoritmo de detecção utilize outras características na sua *pipeline*, como cores predominantes no rótulo por exemplo.

Além disso, a categorização das cervejas por meio de uma Rede Neural ou outro algoritmo de categorização pode ser considerada como alternativa, porém, é necessário avaliar a viabilidade da implementação do mecanismo de detecção utilizando classificadores. Dessa forma, foi desenvolvido uma metodologia para a realização de testes em redes neurais para comprovar se ela seria ideal para o protótipo proposto, para a metodologia foi selecionada a plataforma Tensorflow para o desenvolvimento de uma rede neural convolucional (CNN) de forma que testes possam ser realizados.

O TensorFlow é uma plataforma de inteligência artificial open source mantida e desenvolvida pela Google junto com a comunidade, ela visa o desenvolvimento de algoritmos de *Machine Learning*, possibilitando a utilização modelos em Python, Javascript através do *Web Browser* ou NodeJS, dispositivos mobile e de IoT, entre outros. O desenvolvimento da plataforma em 2018 contava com contribuições de cerca de 400 funcionários da Google e mais de 100 da comunidade (GETTING... [2018]).

Dessa forma, durante os testes seria treinado um modelo de CNN com imagens na resolução 136x76 para posterior classificação em 4 cervejas na camada de saída: Corona, Destroyer Pilsen, Eisenbahn Strong Golden Ale e Apa San Martin.

Figura 18 – Modelo de CNN



Fonte: Do autor.

O modelo de rede neural convolucional proposto possui 6 camadas, conforme podemos visualizar na figura 18, sendo a segunda e terceira camada para a detecção de formas através de camadas convolucionais Conv2D. Porém, para avaliar a dependência da precisão da rede neural no treinamento do modelo, decidiu-se realizar a implementação e testes em duas etapas, treinamento e teste com poucas e muitas imagens exemplares.

8.1.2 Comparações entre plataformas de visão computacional

Uma plataforma de visão computacional pode disponibilizar para o usuário a extração de diversas características de uma foto, entre elas, o texto através do processo de OCR, porém, dependendo da plataforma utilizada, de seu modelo de rede neural ou treinamento realizado no modelo, cada resultado pode ter uma precisão diferente.

Dessa forma, para a escolha provedor de visão computacional preciso o suficiente para o mecanismo de detecção do protótipo, seriam necessários testes entre as 4 plataformas pesquisadas: IBM Watson, Amazon Rekognition, Google Cloud Vision e Microsoft Azure.

8.1.2.1 Metodologia para comparação de plataformas de visão computacional

Para a metodologia, chegou se a conclusão que os testes deveriam conter fotos dos rótulos em condições prováveis para o usuário. Assim sendo, foram escolhidas 5 cervejas com características distintas nos seus rótulos. Conforme pode ser visualizado na figura 19, as cervejas selecionadas e suas características são as seguintes:

- a) Corona: Possui um rótulo impresso no vidro da garrafa.
- b) Destroyer Pilsen: O fundo dos caracteres do rótulo é próximo de branco com caracteres alinhados em fonte simples.
- c) Eisenbahn Strong Golden Ale: Possui um nome alemão com a tradução para 'ferro', o que pode ser um problema para o OCR caso o provedor não possua um dicionário de palavras em outras línguas.

- d) Itaipava Pilsen: Possui caracteres inclinados e com silhueta irregular, além disso, a palavra Itaipava tem origem indígena.
- e) San Martin APA: Alguns caracteres estão ocultos nas bordas por objetos ou outras palavras.

Figura 19 – Cervejas para testes em plataformas



Fonte: Do autor.

Dessa forma, conforme o exemplo da imagem 20, para cada uma das cervejas foram separadas 4 fotos para simular as possíveis situações em que o usuário pode se encontrar no momento que enviar a foto: imagem que possua a mão do usuário segurando a cerveja; capturada em ambiente escuro; foto com o rótulo em destaque; e por último, foto onde a garrafa esteja inclinada ou girando, esta possui o intuito de avaliar o reconhecimento no momento que o usuário não alinha a garrafa corretamente.

Figura 20 – Variações de imagem para testes em plataformas de CV



Fonte: Do autor.

Para os testes, foram utilizadas as páginas de demonstração de funcionalidades que todas as plataformas possuem, em cada uma é possível enviar imagens para que a plataforma demonstre qual o resultado que o consumidor teria ao utilizar a plataforma. Dessa forma, para que seja possível observar os resultados dos testes, foi desenvolvida uma fórmula (2) simples de percentual para avaliar o quanto preciso foi o resultado obtido por cada plataforma. A fórmula deverá ser aplicada após a contagem de caracteres incorretamente detectados de cada palavra serem identificados e somados, onde o C seria a quantidade de caracteres de cada rótulo e I a quantidade de caracteres incorretos somada.

$$\text{Percentual de precisão} = \left(\frac{C}{C-I} \right) * 100 \quad (2)$$

Na tabela 1 podemos observar cada cerveja e as palavras que podem ser consideradas importantes do seu rótulo, assim como a quantidade de caracteres total, valores que serão utilizados para a aplicação da fórmula (2) mencionada.

Tabela 1 – Palavras importantes para comparação de plataformas de CV

| Cerveja | Palavras Importantes | Quantidade de caracteres |
|----------------|----------------------------------|---------------------------------|
| Corona | Corona; Extra | 11 |
| Destroyer | Destroyer; Beer; Pilsen | 19 |
| Eisenbahn | Eisenbahn; Strong; Golden; Ale | 24 |
| Itaipava | Itaipava | 8 |
| Apa San Martin | Barco; Brewers; Apa; San; Martin | 24 |

Fonte: Do autor.

8.1.3 Estrutura do protótipo da API

Para a construção da API, foi definida programação do protótipo através da linguagem Javascript utilizando o ambiente de execução NodeJS, porém, para melhorar a organização e velocidade de desenvolvimento do protótipo, optou-se pela utilização de um framework já existente, dessa forma, foi verificado em testes que a framework TsED continha a maturidade necessária para o projeto.

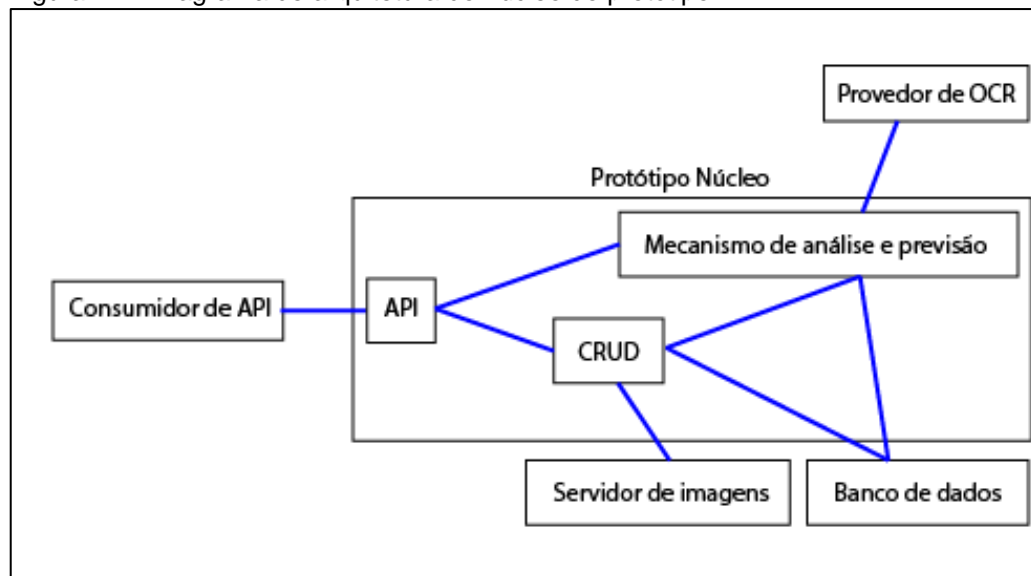
TsED foi criada para facilitar o desenvolvimento de aplicações escritas em NodeJS utilizando Typescript, ela disponibiliza diversas soluções para problemas

encontrados por desenvolvedores durante o desenvolvimento de protótipos e aplicativos para produção, isso é feito através do uso de decoradores, *middlewares* e documentação detalhada para a sua implementação. Além disso, a framework ainda conta com integrações com geradores de documentação, o que facilita a escalabilidade do time de desenvolvedores para posterior transição do protótipo para aplicação comercial.

8.1.4 Arquitetura do núcleo do protótipo

O acesso as funcionalidades do núcleo do protótipo, como cadastro e detecção de rótulos de cervejas, será totalmente disponibilizada através de uma API REST, dessa forma, a API poderá realizar a comunicação com os demais serviços do núcleo para disponibilização para o consumidor.

Figura 21 – Diagrama de arquitetura do núcleo do protótipo



Fonte: Do autor.

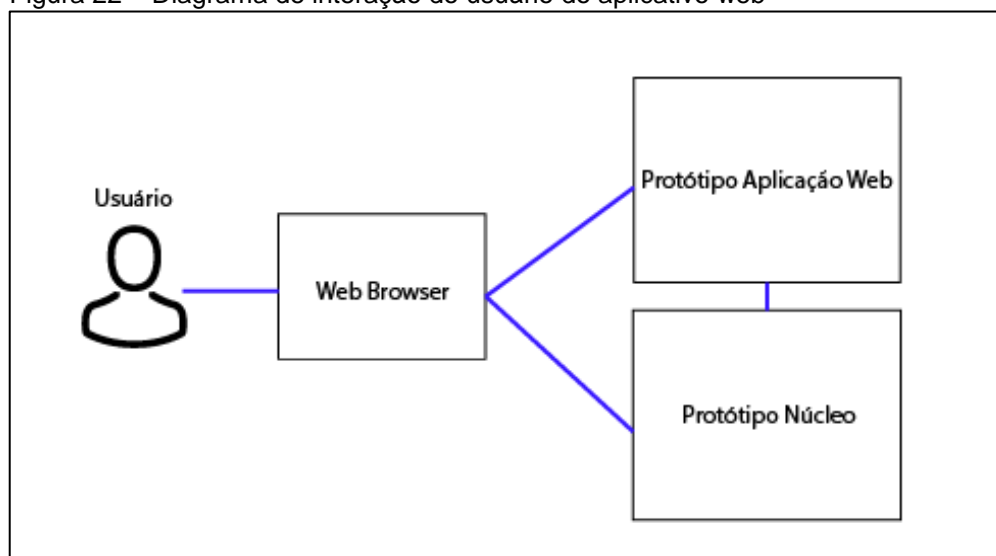
Conforme podemos verificar na figura 21, o único serviço que terá acesso a plataforma do provedor de visão computacional deverá ser o Mecanismo de análise e previsão, este que será responsável tanto por preparar as informações sobre o rótulo para a utilização pelo serviço de leitura, criação, atualização e deleção do banco (CRUD), mas também pela detecção de rótulo que o consumidor de API desejar informações ao utilizar o *endpoint* de previsão na API REST.

Além disso, o Mecanismo de análise e previsão funciona de maneira relativamente simples. Primeiramente, serão armazenadas em memória todas as palavras de todos os rótulos armazenados. Posteriormente, quando o usuário solicitar a detecção de um rótulo, a imagem será enviada para a plataforma de visão computacional para o OCR, com isso, após as palavras do rótulo enviado serem obtidas, elas serão comparadas com a lista de palavras de cada cerveja já armazenada em banco. Dessa forma, será possível gerar um nota de similaridade para cada cerveja e eleger como cerveja detectada a com maior nota dentre todas.

8.1.5 Arquitetura da aplicação web do protótipo

Após a conclusão do protótipo do núcleo, de forma que seja possível demonstrar as funcionalidades presentes, optou-se por desenvolver um aplicativo web independente que consuma as APIs disponibilizadas. Dessa forma, conforme observado na figura 22, ao acessar o aplicativo utilizando um *web browser*, o usuário iniciaria uma sessão tanto com o servidor do protótipo de aplicação web, que serviria para o disponibilizar as telas do aplicativo, quanto o do núcleo, que iria disponibilizar as informações para demonstração nas telas acessadas. Além disso, a comunicação entre os dois servidores, aplicação web e núcleo, seria possível para eventuais consultas e validações necessárias.

Figura 22 – Diagrama de interação do usuário do aplicativo web



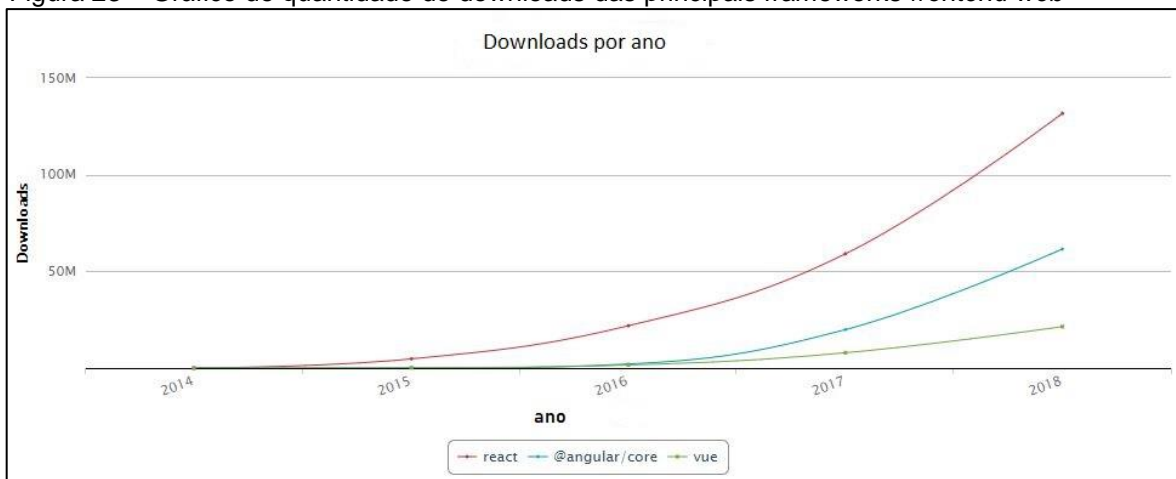
Fonte: Do autor.

Dessa forma, optou-se pela utilização de NodeJS no também no protótipo de aplicação web, este que seria responsável por montar uma sessão do usuário, realizar validações de acesso a rotas com o protótipo de núcleo e servir os documentos HTML e outros para o navegador de internet do usuário. Além disso, para a criação de telas de aplicativo web, foi escolhida a biblioteca ReactJS para proporcionar interações ricas e dinâmicas para o usuário através do navegador utilizando HTML, CSS e Javascript.

8.1.5.1 ReactJS

ReactJS é um biblioteca de Javascript para a criação de interfaces de usuários. A biblioteca hoje é uma das escolhas mais utilizadas por desenvolvedores *frontend*, conforme podemos observar na imagem 23. Além disso, ela vem sendo desenvolvida em conjunto com a comunidade de desenvolvedores contribuintes e também do Facebook, este que criou a biblioteca e a utiliza até hoje nos seus principais aplicativos, porém, devido a licença open source do ReactJS, a framework atraiu diversos desenvolvedores ao redor do mundo, estes que utilizam e contribuem para o crescimento da framework.

Figura 23 – Gráfico de quantidade de downloads das principais frameworks *frontend web*



Fonte: Adaptado de Npm (2019).

Uma das principais vantagens em utilizar um framework como ReactJS é a possibilidade de segmentação do código em diversos componentes, dessa forma, a

definição de um conjunto de tags HTML e suas logicas de interação podem ser reutilizadas em diversas partes do aplicativo.

Figura 24 – Criação de componentes em ReactJS

```
class ShoppingList extends React.Component {  
  render() {  
    return (  
      <div className="shopping-list">  
        <h1>Shopping List for {this.props.name}</h1>  
        <ul>  
          <li>Instagram</li>  
          <li>WhatsApp</li>  
          <li>Oculus</li>  
        </ul>  
      </div>  
    );  
  }  
}
```

// Example usage: <ShoppingList name="Mark" />

Fonte: "Reactjs.org" ([2019]).

Assim, conforme demonstrado na imagem 24, é possível criar e reutilizar um componente através do Javascript, este que utiliza tags de marcação dentro do próprio script do componente, isso é possível através dos arquivos com extensão JSX, que posteriormente são compilados para um script padrão que adiciona as tags necessárias no documento HTML visualizado pelo usuário. Dessa forma, o componente poderia se comportar diferentemente em locais distintos do aplicativo, onde seu comportamento depende de seu estado interno ou dos dados foram disponibilizados para o componente durante a sua inicialização. Além disso, ReactJS possui diversas outras funcionalidades e bibliotecas robustas que permitem a criação de uma aplicação completa, onde alguns exemplos seriam os produtos Facebook, Instagram, Netflix, Dropbox, etc.

8.2 RESULTADOS

O objetivo do trabalho proposto foi alcançado através da construção de um protótipo de aplicativo para catálogo de cervejas, este que possui a capacidade de reconhecer rótulos de cervejas para melhorar a experiência do usuário na busca do produto desejado. Dessa forma, para a criação do mecanismo de detecção, primeiramente foi comprovada que a extração de texto seria a melhor alternativa para o caso de uso, para isto, foi realizado estudos e teste de métodos alternativos. Após a comprovação que o OCR seria a melhor alternativa, foram realizados testes em 4 plataforma de visão computacional disponíveis, dessa forma, a melhor plataforma pode ser identificada para utilização. Assim sendo, a construção do protótipo foi concluída conforme a arquitetura proposta na metodologia, assim também como a inclusão de informações no banco conforme métodos de extração de dados da *web*.

8.2.1 Métodos alternativos ao OCR para reconhecimento de rótulos

Conforme a metodologia proposta, foram realizados testes em uma CCN para verificação da viabilidade de sua utilização no mecanismo de detecção no núcleo do protótipo. Dessa forma, para o primeiro teste com poucos exemplares, foram escolhidas apenas 4 imagens para treinamento, 1 de cada rótulo de cerveja, e conforme pode ser visualizado na figura 25, foram utilizadas imagens convertidas em escala de cinza para simplificação do modelo.

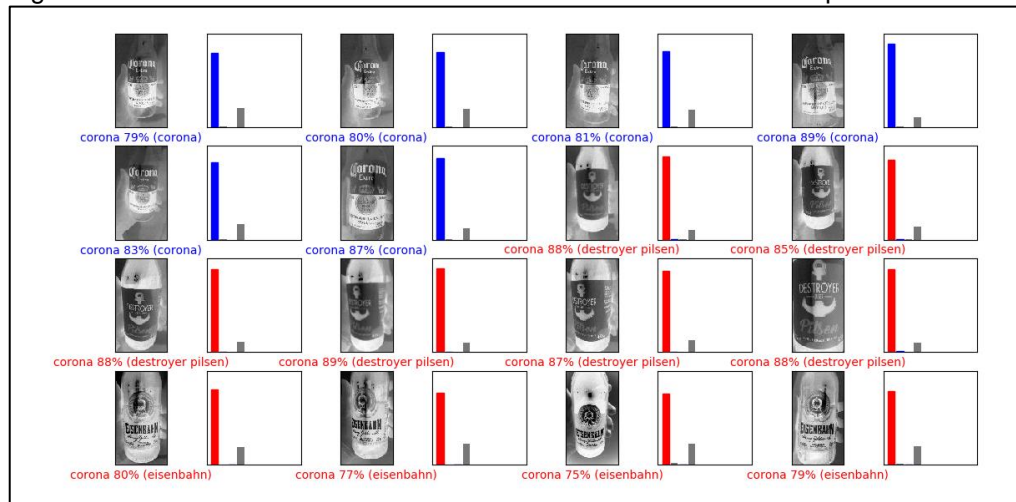
Figura 25 – Quatro rótulos para treinamento de CNN



Fonte: Do autor.

Após treinar o modelo por 5 *epochs*, foram utilizadas 16 imagens, 4 de cada rótulo de cerveja para testar a taxa de acerto da rede neural. Conforme pode ser visualizado na figura 26, os resultados em vermelho indicam uma detecção incorreta, dessa forma, pode ser verificado que a rede obteve uma taxa de precisão pequena, sendo que os acertos ocorreram pois a rede classificou todos os rótulos como pertencentes a cerveja Corona.

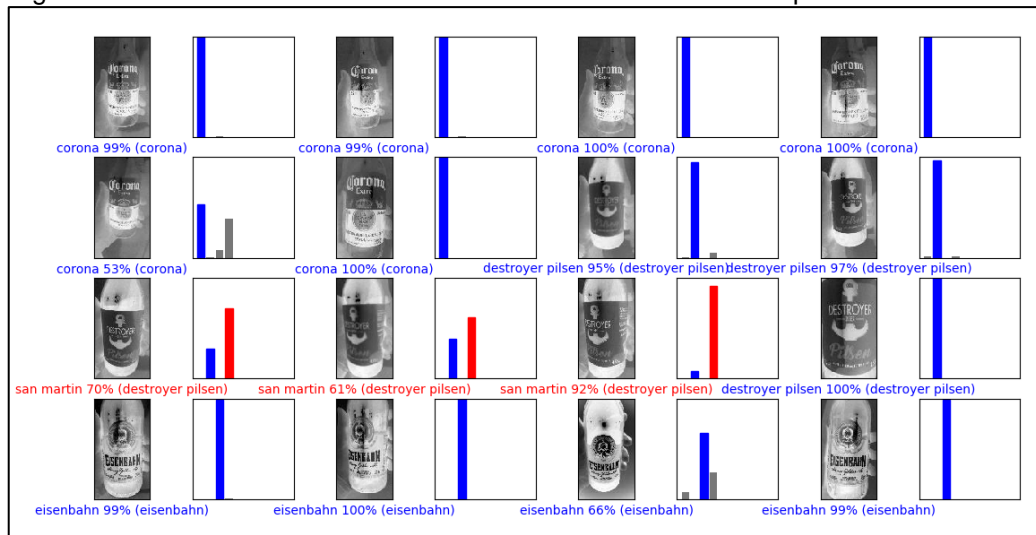
Figura 26 – Resultado de testes de CNN com baixo número de exemplos



Fonte: Do autor.

Dessa forma, foi necessário a realização de um segundo teste, desta vez com 280 de rótulos divididos igualmente entre as 4 cervejas, após o treinamento, foram realizados novamente os teste com os 16 rótulos do exemplo anterior.

Figura 27 – Resultado de teste de CNN com alto número de exemplos



Fonte: Do autor.

Como pode ser observado na figura 27, ao aumentarmos o número de exemplos de 4 para 280, a taxa de erro foi de 62% para 18%, o que confirma a alta dependência de um sistema classificador com CNN na quantidade de exemplos durante o seu treinamento, porém, isso pode ser identificado na literatura como uma característica de redes neurais. Dessa forma, caso a inclusão de novas cervejas no banco de dados ser realizado por o consumidor do aplicativo, a dependência em exemplares por parte dos classificadores em CNN se torna uma barreira para o mecanismo de detecção de rótulos ser implementado, isso devido a necessidade de o usuário incluir diversas fotos do mesmo rótulo, que tornaria dificultosa a utilização do aplicativo comercialmente.

8.2.2 Comparações entre plataformas de visão computacional

Os testes foram realizados conforme a metodologia proposta, os resultados finais podem ser observados na tabela 2.

Tabela 2 – Resultado de testes entre plataformas de CV

| Cerveja | Situação | Plataforma | % Acerto |
|-----------|----------------------|---------------------|----------|
| Corona | Com mão | IBM Watson | 45,5 |
| | | Amazon Rekognition | 72,7 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 100 |
| | | | |
| | Pouca Luz | IBM Watson | 18,2 |
| | | Amazon Rekognition | 63,6 |
| | | Google Cloud Vision | 90,9 |
| | | Microsoft Azure | 100 |
| | | | |
| | Rótulo apenas | IBM Watson | 81,8 |
| | | Amazon Rekognition | 72,7 |
| | | Google Cloud Vision | 81,8 |
| | | Microsoft Azure | 63,6 |
| | | | |
| | Inclinado ou girando | IBM Watson | 72,7 |
| | | Amazon Rekognition | 90,9 |
| | | Google Cloud Vision | 90,9 |
| | | Microsoft Azure | 90,9 |
| | | | |
| Destroyer | Com mão | IBM Watson | 84,2 |

| | | | |
|-----------|----------------------|---------------------|------|
| Eisenbahn | | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 100 |
| | Pouca Luz | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 100 |
| | Rótulo apenas | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 78,9 |
| | Inclinado ou girando | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 94,7 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 50 |
| Itaipava | Com mão | Amazon Rekognition | 75 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 95,8 |
| | | IBM Watson | 25 |
| | Pouca Luz | Amazon Rekognition | 75 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 95,8 |
| | | IBM Watson | 50 |
| | Rótulo apenas | Amazon Rekognition | 79,2 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 95,8 |
| | | IBM Watson | 33,3 |
| | Inclinado ou girando | Amazon Rekognition | 75 |
| | | Google Cloud Vision | 50 |
| | | Microsoft Azure | 83,3 |
| | | IBM Watson | 37,5 |
| | Com mão | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 0 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 37,5 |
| | Pouca Luz | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 0 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 37,5 |
| | Rótulo apenas | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 0 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 37,5 |

| | | | |
|----------------|----------------------|---------------------|------|
| Apa San Martin | Inclinado ou girando | Amazon Rekognition | 100 |
| | | Google Cloud Vision | 100 |
| | | Microsoft Azure | 100 |
| | | IBM Watson | 37,5 |
| | | Amazon Rekognition | 87,5 |
| | | Google Cloud Vision | 87,5 |
| | | Microsoft Azure | 87,5 |
| | | IBM Watson | 70,8 |
| | Com mão | Amazon Rekognition | 95,8 |
| | | Google Cloud Vision | 50 |
| | | Microsoft Azure | 87,5 |
| | | IBM Watson | 62,5 |
| | Pouca Luz | Amazon Rekognition | 95,8 |
| | | Google Cloud Vision | 50 |
| | | Microsoft Azure | 83,3 |
| | | IBM Watson | 62,5 |
| | Rótulo apenas | Amazon Rekognition | 91,7 |
| | | Google Cloud Vision | 50 |
| | | Microsoft Azure | 62,5 |
| | | IBM Watson | 62,5 |
| | Inclinado ou girando | Amazon Rekognition | 79,2 |
| | | Google Cloud Vision | 50 |
| | | Microsoft Azure | 75 |
| | | IBM Watson | 70,8 |

Fonte: Do autor.

Dessa forma, podemos observar no gráfico de percentual de precisão da figura 28 que, dentre todas as plataformas, a IBM Watson foi a que menos obteve precisão no OCR, seguido por Google Cloud Vision.

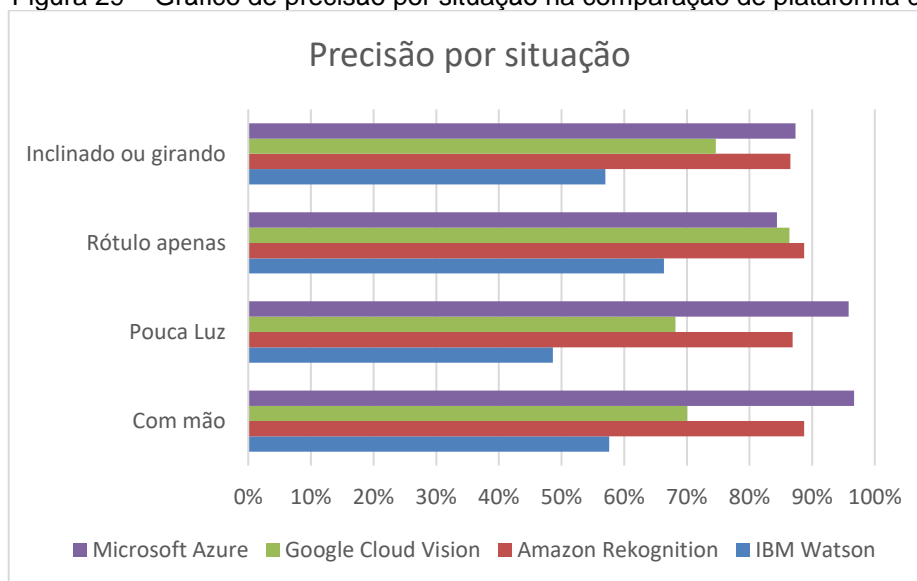
Figura 28 – Gráfico de precisão geral na comparação de plataforma de CV



Fonte: Do autor.

Ao realizar a análise com medição da precisão por situação, conforme a figura 29, verificou-se também que as plataformas IBM Watson e Google Cloud Vision possuem uma dificuldade com imagens escuras, onde alcançaram 49% e 68% de precisão em seu OCR nessa ordem, diferente de Microsoft Azure ou Amazon Rekognition, que obtiveram uma precisão muito similar ou até mesmo superior ao escurecer as imagens.

Figura 29 – Gráfico de precisão por situação na comparação de plataforma de CV



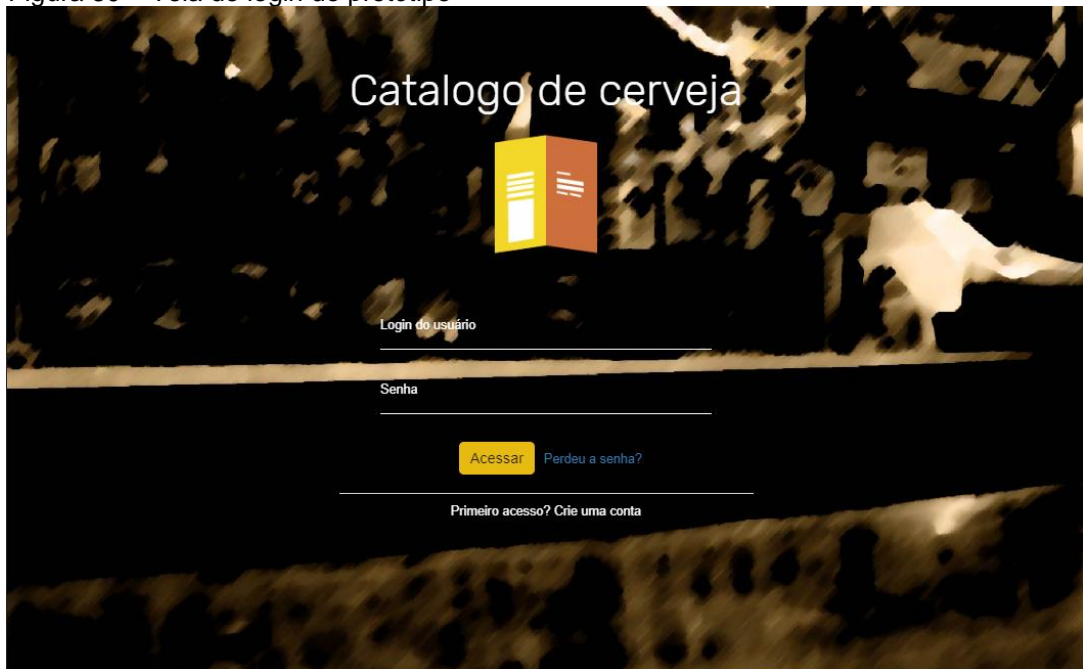
Fonte: Do autor.

Além disso, para as plataformas Google Cloud Vision e IBM Watson a técnica de enviar uma imagem apenas o rótulo recortado da cerveja contribuiu para um aumento significativo na precisão, ao contrário de Microsoft Azure, que apresentou um resultado ainda pior. Porém, por hora pode-se descartar o uso das plataformas IBM Watson e Google Cloud Vision para a utilização no protótipo final, isso porque ambas dependeriam do desenvolvimento de um mecanismo auxiliar de recorte de rótulo.

8.2.3 Aplicação

Após o desenvolvimento da aplicação, obtivesse uma interface com tela de login, cadastro de usuário e catalogo de cervejas. A tela de login, conforme demonstrado na figura 30, possui também os atalhos para o cadastro do usuário.

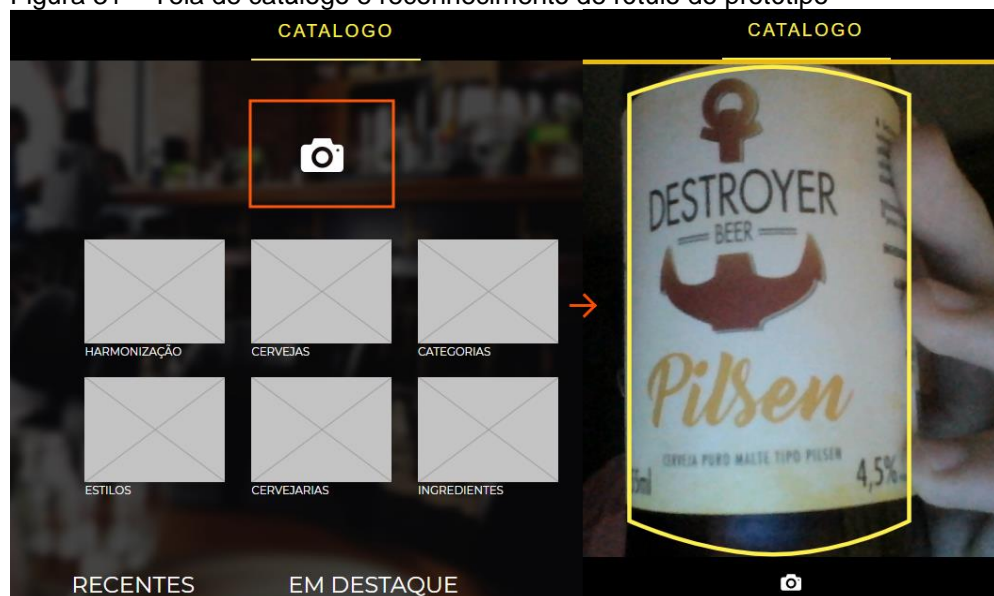
Figura 30 – Tela de login do protótipo



Fonte: Do autor.

Após o login, é demonstrado para o usuário o catalogo de cervejas, o usuário pode optar em realizar a consulta de cervejas por texto utilizando o nome da cerveja, estilo, categorias, entre outros, porém, para facilitar a identificação, é possível utilizar também o mecanismo de análise e previsão através da câmera do dispositivo, conforme demonstrado na figura 31.

Figura 31 – Tela de catálogo e reconhecimento de rótulo do protótipo



Fonte: Do autor.

Assim que a cerveja desejada ser encontrada, um cadastro detalhado é demonstrado para o usuário, este que ainda pode avaliar a cerveja e comentar a sua avaliação, conforme a figura 32.

Figura 32 – Tela de cadastro de cerveja do protótipo



Fonte: Do autor.

Dessa forma, o usuário consegue interagir com os recursos disponíveis no núcleo do protótipo e realiza suas consultas assistido pela visão computacional utilizando o OCR dos rótulos, sem a necessidade de inserir nenhuma informação para que o aplicativo a identifique a cerveja. Porém, para comprovar a utilidade e precisão do mecanismo de detecção, com apenas três cervejas inclusas no banco, foram realizados três testes de reconhecimentos com cada uma, onde entre todas os testes, em somente uma das análises houve um retorno incorreto. Assim, pode ser concluído que, com poucas cervejas no banco e com um algoritmo simples no mecanismo de análise, o protótipo já apresenta uma precisão de aceitável de 88%, o que seria superior ao obtido anteriormente no trabalho de Silva (2014) ao utilizar o motor Tesseract e buscar informações no motor de busca do Google, porém, com a desvantagem de detectar somente cervejas que foram cadastradas anteriores a detecção.

8.2.4 Extração de dados para composição de banco

Para futura realização de testes em situação mais realistas, é necessário uma quantia significativa de dados para a composição de um banco de dados cervejeiro, dessa forma, primeiramente foi levantado dados referentes a alguns sites com informações suficientes disponíveis sobre cervejas. Porém, conforme necessidade do mecanismo de detecção, a fonte de dados deveria conter também imagens dos rótulos das cervejas, dessa forma, seria possível realizar o OCR do rótulo para inclusão no banco e posterior reconhecimento. Dentre os sites encontrados estão:

- a) emporiodacerveja.com.br: site brasileiro com informações de cervejas tipicamente consumidas no território brasileiro, como não possui API para consulta, uma extração de dados utilizando *Web Scraping* é necessária.
- b) clubedomalte.com.br: outro site brasileiro disponível, possui algumas informações particulares sobre cervejas, porém, as informações estão de menos acordo com os padrões internacionais, não possui API.
- c) brewerydb.com: banco de dados internacional de cervejas, possui uma API e também imagens recortadas dos rótulos das cervejas, o que facilita a extração e preparação de dados para o banco.

Dessa forma, devido a facilidade de extração e conter as imagens dos rótulos já preparadas, Brewerydb.com foi escolhida como fonte de dados para temporariamente compor o banco do protótipo obtido. Um simples script escrito com NodeJS 10 foi desenvolvido para a realização de requisições a API disponível, após, os dados das cervejas obtidos foram armazenados no formato do JSON em arquivos no disco. Conforme o anexo B, o modelo JSON obtido contém informações detalhadas sobre cervejaria, estilo, categoria, etc. Os arquivos salvos em disco posteriormente foram tratados com outro script para gerar comandos SQL, estes que foram utilizados para inserir no banco de dados informações sobre cervejas, cervejarias, estilos e categorias de cervejas extraídas.

Assim, foi possível obter dados completos de cerca de 2500 cervejas com seus respectivos rótulos, dessa forma, posteriormente, foram enviadas todas as imagens para a API da plataforma Amazon Rekognition, onde foi possível obter

também o texto extraídos de todos os rótulos. Porém, em razão da necessidade de criação de um script para consolidação das informações, dentro do cronograma disponível não foi possível realizar a inclusão das informações extraídas no banco de dados para testes.

9 CONCLUSÃO

Através das pesquisas realizadas para o referencial teórico e das técnicas descobertas durante a criação do protótipo, foi possível compreender as mais variadas possibilidades de aplicações que a visão computacional proporciona. Além disso, foi possível compreender diversos obstáculos que ainda existem para que todos os itens do dia a dia sejam identificáveis por uma câmera de computador. Porém, de acordo com o que foi demonstrado no projeto, é possível contornar alguns desses problemas para que a cada dia o computador possa auxiliar os usuários em cada vez mais tarefas que podem ser consideradas simples, como busca de um produto em um catálogo por exemplo.

O conhecimento adquirido sobre redes neurais e a aplicação em alguns modelos de testes necessários esclareceu diversas dúvidas sobre o funcionamento de sistemas de visão computacional, permitindo que ocorra futura aplicação desses conhecimentos em modelos que melhorem os resultados obtidos.

Os temas não pertencentes a inteligência artificial, como os modelos para extração de dados da web, possibilitou uma melhor compreensão da importância de um fonte de dados confiável e de fácil acesso para o desenvolvimento de protótipos ou aplicativos. Além disso, o estudo de IoT possibilitou uma melhor visão do possível futuro da computação e da interação das coisas presentes na rotina humana, possibilitando que futuramente, uma melhor metodologia para detecção de produtos com rótulos em nosso redor possa ser pensada e desenvolvida.

Através das pesquisas e desenvolvimento do protótipo, foi possível identificar algumas das possíveis arquiteturas que um sistema de detecção e detalhamento de rótulos pode ter, como o modelo observado no trabalho de Silva (2014), além de realizar a comparação com a arquitetura escolhida para este trabalho. Além disso, foram descobertas novas tecnologias de desenvolvimento de aplicações, além da compreensão de como projetar uma arquitetura que possa posteriormente ser integrada a outros produtos, permitindo o crescimento do ecossistema de soluções, que somadas, podem agregar em uma aplicação útil para a sociedade. Dessa forma, foi possível encontrar a solução para a realização do OCR dos rótulos, que caso não integradas e utilizadas as plataformas de visão computacional disponibilizadas em nuvem conforme foi feito durante o trabalho, iriam tomar um tempo

de desenvolvimento possivelmente maior e com menor taxa de precisão. Além disso, foi possível entender as características de redes neurais como classificadores que tornaria não viável a sua aplicação em um mecanismo de detecção caso as novas informações de banco de dados serem inseridas pelo consumidor final.

Porém, foi identificado durante os testes com o protótipo final que existem diversas limitações que devem ser tratadas antes de um sistema com a mesma arquitetura ser disponibilizado em produção, como por exemplo, a identificação incorreta de um único carácter de uma palavra, o que pode gerar um resultado diferente do desejado na detecção. Dessa forma, uma possível solução a ser implementada futuramente seria um algoritmo para considerar similaridade das palavras sendo comparadas, assim, o impacto de um carácter incorretamente detectado poderia ser consideravelmente menor.

Outro ponto que poderia ser mais explorado seria testar a escalabilidade do algoritmo de detecção ao utilizar diversos cadastros de produtos, isso porque, por ser um algoritmo simples, pode gerar lentidões caso não otimizado.

Além disso, para utilização de um sistema similar em produção, será necessário pagar o custo da plataforma de visão computacional escolhida. Dessa forma, poderia ser estudado e implementado um sistema de OCR próprio, este que permitiria a diminuição do custo total e possivelmente facilitaria uma melhora na precisão dos caracteres, já que poderiam ser adicionadas fontes comuns nos rótulos de cervejas durante o treinamento dos modelos de rede neurais.

Por fim, um modelo de CNN para auxílio de um sistema para recorte dos rótulos nas imagens disponibilizadas também poderia ser desenvolvido em um trabalho futuro, este que, dependendo da plataforma escolhida, irá melhorar a precisão do OCR. Além disso, tal sistema iria tornar as fotos dos rótulos no cadastro do produto organizadas, já que seria demonstrado apenas o rótulo do produto consultado na imagem do produto.

REFERÊNCIAS

AL-FUQAHA, Ala et al. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. **Ieee Communication Surveys & Tutorials**. Nova Jersey, p. 2347-2376. Out. 2015. Disponível em: <<https://ieeexplore.ieee.org/document/7123563>>. Acesso em: 14 out. 2018.

AMAZON. **Amazon Rekognition**. [2018]. Disponível em: <<https://aws.amazon.com/pt/rekognition/>>. Acesso em: 27 nov. 2018.

ANDRADE FILHO, Eládio Pessoa de; PEREIRA, Francisco Carlos Ferreira. **Anatomia Geral**. Sobral: Faculdades Inta, 2015. Disponível em: <<http://md.intaead.com.br/geral/anatomia-geral/pdf/anatomia-geral.pdf>>. Acesso em: 01 out. 2018.

ARASU, Arvind; GARCIA-MOLINA, Hector. **Extracting structured data from Web pages**. Proceedings of the 2003 Acm Sigmod International Conference on Management Of Data - Sigmod '03, [s.l.], p.337-348, 2003. ACM Press. <<http://dx.doi.org/10.1145/872757.872799>>. Disponível em: <<https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/extract.pdf>>. Acesso em: 14 out. 2018.

BATISTA, Miguel João Rodrigues. **Sistema de Digitalização, Detecção e Classificação Automática de Recibos em Dispositivos Móveis**. 2012. 81 f. Dissertação (Mestrado) - Curso de Engenharia de Redes de Comunicação e Multimédia, Instituto Superior de Engenharia de Lisboa, Lisboa, 2012. Disponível em: <<https://repositorio.ipl.pt/bitstream/10400.21/2135/1/Disserta%C3%A7%C3%A3o.pdf>>. Acesso em: 18 nov. 2018.

BIEHL, Matthias. **RESTful API Design: APIs your consumers will love**. [s. L.]: Createspace Independent Publishing Platform, 2016.

BRAGA, Antônio de Pádua; CARVALHO, André Ponce de Leon F. de; LUDEMIR, Teresa Bernarda. **Redes Neurais Artificiais: Teoria e aplicações**. Rio de Janeiro: Ltc - Livros Técnicos e Científicos, 2000

CERAMI, Ethan. **Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL**. Publisher: O'Reilly, 2002.

CORDING, Patrick. (2018). **Algorithms for Web Scraping**.

D'ALBE, E. E. F.. On a Type-Reading Optophone. **Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences**, [s.l.], v. 90, n. 619, p.373-375, 1 jul. 1914. The Royal Society. Disponível em: <<http://dx.doi.org/10.1098/rspa.1914.0061>>. Acesso em: 15 set. 2018.

DUCKETT, Jon. **HTML & CSS: Design and build Websites**. Indianapolis, Eua: John Wiley & Sons, Inc., 2011.

EVANS, Dave. **A Internet das coisas**: como a próxima evolução da internet está mudando tudo. 2011. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_0411final.pdf>. Acesso em: 02 jul. 2018.

GETTING Started with TensorFlow and Deep Learning. [s.i]: Enthought, [2018]. P&B. Disponível em: <<https://www.youtube.com/watch?v=tYYVSEHq-io&t=496s>>. Acesso em: 12 jun. 2019.

GITHUB.COM/MICHADA. **Create a RSS feed from www.kickstarter.com**. Disponível em: <<https://sing-group.org/jarvest/examples.html>>. Acesso em: 02 set. 2018.

GLEZ-PENÑA, Daniel et al. Web scraping technologies in an API world. **Briefings In Bioinformatics**, Oxford, v. 15, n. 5, p.788-797, 1 set. 2014.

GNU.ORG. **The GNU Awk User's Guide**. Disponível em: <<https://www.gnu.org/software/gawk/manual/gawk.html>>. Acesso em: 14 out. 2018.

GONÇALVES, Julio Cesar. **Reconhecimento de dígitos em imagens de medidores de consumo de gás natural utilizando técnicas de visão computacional**. 2016. 119 f. Dissertação (Mestrado) - Curso de Engenharia de Sistemas Computacionais, Universidade Tecnológica Federal do Paraná, Curitiba, 2016. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/2210/1/CT_PPGCA_M_Gon%C3%A7alves%20Julio%20Cesar_2016.pdf>. Acesso em: 18 nov. 2018.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [s. L.]: Mit Press, 2016. Disponível em: <<http://www.deeplearningbook.org>>. Acesso em: 04 jun. 2019.

GOOGLE. **Cloud Vision**. [2018]. Disponível em: <<https://cloud.google.com/vision/>>. Acesso em: 27 nov. 2018.

GSMA. **Number of global mobile subscribers to surpass five billion this year, finds new GSMA study**. 2017. Disponível em:

<<https://www.gsma.com/newsroom/press-release/number-of-global-mobile-subscribers-to-surpass-five-billion-this-year/>>. Acesso em: 21 nov. 2018.

HAYKIN, Simon. **Neural Networks and Learning Machines**. 3. ed. Hamilton: Prentice Hall, 2009.

IBM. **Why Watson is AI for business**. [2018]. Disponível em: <<https://www.ibm.com/watson/>>. Acesso em: 27 nov. 2018.

HEATON, Jeff. **Programming Spiders, Bots and Aggregators in Java**. Nova York: John Wiley & Sons Inc, 2002.

INTERNATIONAL TELECOMMUNICATION UNION. **ITU Internet Reports 2005: The Internet of Things**. Tunísia, 2015. 28 p. Disponível em: <https://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf>. Acesso em: 08 out. 2018.

INTRODUCING our advanced features. 2018. Disponível em: <<https://dexi.io/>>. Acesso em: 09 set. 2018.

KALIN, Martin. **Java Web Services: Up and Running: A Quick, Practical, and Thorough Introduction**. 2. ed. Sebastópolis: O'reilly Media, 2013.

LUDWIG JUNIOR, Oswaldo; COSTA, Eduard Montgomery Meira. **Redes Neurais: Fundamentos e Aplicações com programas em C**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

LUGER, George F.. **Inteligência artificial**. 6. ed. São Paulo: Pearson Education do Brasil Ltda., 2013. 614 p. Tradução de Daniel Vieira.

MCEWEN, Adrian; CASSIMALLY, Hakim. **Designing the Internet of Things**. Chichester: John Wiley & Sons, 2014.

MENDONÇA, Fábio Lúcio Lopes de. **Proposta de arquitetura de um sistema com base em OCR neuronal para resgate e indexação de escritas paleográficas do SEC.XVI ao XIX**. 2008. 118 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade de Brasília, Brasília, 2008. Disponível em: <http://repositorio.unb.br/bitstream/10482/1157/1/DISSERTACAO_2008_FabioLucioLMendonca.pdf>. Acesso em: 18 nov. 2018

MICROSOFT. **Computer Vision**. Disponível em: <<https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>>. Acesso em: 04 nov. 2018.

MITCHELL, Ryan. **Web Scraping with Python**. Sebastopol: O'reilly, 2015. 255 p.

MOBILEYE. **Our Technology**. [20--]. Disponível em: <<https://www.mobileye.com/our-technology/>>. Acesso em: 12 jun. 2019.

NETO, Luiz Garcia Palma; NICOLETTI, Maria do Carmo. **Introdução às Redes Neurais Construtivas**. São Carlos: Edufscar, 2005.

NPM. **Download statistics for packages react, vue, @angular/core**. 2019. Disponível em: <<https://npm-stat.com/charts.html?package=react&package=vue&package=%40angular%2Fcore&from=2014-12-12&to=2018-12-12>>. Acesso em: 12 jun. 2019.

READHEAD, Josh. **Machine Learning**: how support vector machines can be used in trading. 2012. Disponível em: <<https://www.mql5.com/en/articles/584>>. Acesso em: 25 nov. 2018.

READING and Sorting Mail Automatically. [s.i]: Us Postal Service, 1970. Son., color. Disponível em: <<https://www.youtube.com/watch?v=cdfYrpRHqJ0>>. Acesso em: 15 set. 2018.

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. **Redes Neurais Artificiais**: para engenharia e ciências aplicadas. São Paulo: Artliber Editora Ltda., 2010. 399 p.

SILVA, Luana Gomes. **Protótipo de aplicativo Android para extração de texto em imagens para busca semântica sobre rótulos de cervejas**. 2014. 87 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2014.

SNELL, Richard S. **Neuroanatomia clínica**. 7ª. ed. Rio de Janeiro: Editora Guanabara Koogan, 2011

SZELISKI, Richard. **Computer Vision**: Algorithms and Applications. Nova York: Springer, 2011.

VERGHESE, Vijay. **Google Translate “instant visual translation” updated with 20 new languages**. 2015. Disponível em: <<https://www.androidguys.com/news/google-translate-updated-with-20-new-languages/>>. Acesso em: 15 set. 2018.

VON ZUBEN, Prof. Fernando J.. **Redes Neurais Artificiais e Máquinas de Aprendizado (Parte 1)**. Disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ea072_2s15/topico1_EA072_2s2015_Parte1.pdf>. Acesso em: 22 out. 2018.

WORD ASSOCIATION OF NEWS PAPERS. **Newspapers:** 400 Years Young!
Disponível em: <<http://archive.li/L7Z9t>>. Acesso em: 25 set. 2018.

APENDICE(S)

APÊNDICE A – Artigo Científico

Implementação de protótipo baseado na tecnologia OCR aplicada ao reconhecimento de rótulos para busca em banco de dados

David Vitor Antonio¹, Gustavo Bisognin²

1 Acadêmico do Curso de Ciência da Computação – Departamento de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brazil

2 Professor do Curso de Ciência da Computação – Departamento de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brazil

David.vitora@gmail.com, gbsogn@gmail.com

Abstract. *Internet of Things is a concept that aims at connecting all the things that are used in the daily routine to automate and make human duties easier. From Smartphones to umbrellas; refrigerators and other items. It can be seen that the number of objects being connected to a network and communicating with other devices is increasing. However, disposable things like products' packages can also become part of this list of detectable things, thus, it would be possible, for instance, to visualize detailed information about a product using smartphones only by approximating the device with the product's label. However, nowadays, the cost of turning a label detectable can make the product detection inviable. Thus, as an alternative to the described problem, this article shows an implementation of an architecture and prototype that uses optical character recognition (OCR) to do label recognition through the use of an existing computer vision platform. The prototype can be accessed using the web browser of a computer with internet access, so if the user accesses the app and sends a picture of a beer label, the image is sent to a server that communicates with the computer vision platform, later, the returned information will be used to find information about the beer in the database and it will be sent to the user, also allowing the user to rate the product. Furthermore, tests to check if the usage of Convolutional Neural Networks (CNN) as classifier can meet the needs of the detection mechanism were performed, that was possible while studying concepts of neural networks, its architecture and also developing a small CNN. In such a way, it was possible to confirm that it would not be possible to use image classifiers since that in the proposed architecture the user would be responsible for submitting new data and it requires a significant amount of image data to train a CNN.*

Resumo. *A Internet das coisas visa conectar todos as coisas presentes no dia a dia para a facilitar e automatizar a rotina humana. Desde celulares; guarda chuvas; geladeiras e diversos outros itens, cada vez mais pode se observar objetos sendo conectados a uma rede e realizando a comunicação com outros dispositivos. Porém, coisas degradáveis como embalagens de produtos também podem fazer deste conjunto de itens detectáveis, dessa forma, seria possível por exemplo, visualizar informações mais detalhadas sobre um produto no celular apenas ao aproximar o dispositivo com o rótulo. Porém, atualmente, o custo para tornar todos os rótulos detectáveis pode tornar inviável a detecção de produtos descartáveis. Dessa forma, como alternativa ao problema descrito, este trabalho apresenta a implementação de um protótipo de arquitetura e software para que seja realizada a detecção dos rótulos através da câmera do dispositivo utilizado, isso é possível através do reconhecimento ótico de caracteres (OCR), este que pode ser realizado através de uma plataforma de visão computacional já existente. O protótipo pode ser acessado via navegador de internet de um computador, dessa forma, ao acessa-lo e capturar uma foto do rótulo de uma cerveja, a foto é enviado para o servidor, este que realiza o OCR através da plataforma de visão computacional, busca as informações em um banco de dados referentes a cerveja indicada e mostra para o usuário através do navegador, permitindo também que o usuário avalie a cerveja. Além disso, durante o trabalho são realizados testes com redes neurais convolucionais para a avaliação de viabilidade para utiliza-las em um mecanismo de detecção com classificadores, para isto, foram estudados os conceitos de redes neurais e suas arquiteturas, além da implementação de um pequeno protótipo rede neural convolucional (CNN) para teste. Dessa forma, comprovou-se inviabilidade de utilização de classificadores de imagens já que na arquitetura proposta o usuário seria o responsável por inserir novas informações e o treinamento de classificadores com CNN requer uma quantidade significativa de dados de imagens.*

1. Introdução

A “Internet das Coisas” se refere a uma revolução tecnológica que tem como objetivo conectar os itens usados do dia a dia à rede mundial de computadores. Neste contexto, cada vez mais, os dispositivos computacionais se conectam e se comunicam com as coisas do nosso cotidiano. Atualmente se pode observar o surgimento de diversos eletrônicos como eletrodomésticos e meios de transporte que estão conectados à Internet e a outros dispositivos, como computadores e smartphones.

Porém, para a possível quebra das barreiras tecnológicas que geralmente limitam a IoT preferencialmente para dispositivos eletrônicos, é necessário unir as mais diversas áreas da tecnologia, como por exemplo, o campo da Inteligência Artificial (IA). A IA apresentou um crescimento significativo nos últimos anos, proporcionando o desenvolvimento de diversos tipos de algoritmos para realizar processamento similar ao raciocínio humano. Dessa forma, foram encontradas possibilidades de aplicações do raciocínio artificial através da Visão Computacional para permitir a extração de

características de imagens através de algoritmos avançados. Com isso, um sistema de reconhecimento de imagens associado a um banco de dados com informações sobre produtos, poderá realizar uma análise das imagens de rótulos de produtos descartáveis e disponibilizar os dados para o consumidor. Diante dos fatos apresentados, este trabalho propõe a utilização da funcionalidade de reconhecimento de caracteres de imagens presente em plataformas de visão computacional para a criação de um protótipo que poderá realizar a detecção de cervejas através de fotos dos seus rótulos. Assim, o protótipo é capaz de utilizar a cerveja detectada para buscar informações sobre o produto em um banco de dados interno que possui informações sobre o produto. Além disso, para a escolha da plataforma de visão computacional ideal, o trabalho realiza uma comparação de desempenho entre 4 plataformas de visão computacional disponíveis para a utilização.

1. Modelos de extração de dados

Um protótipo que realiza a busca de informações sobre produtos em uma base de dados interna necessita de informações para compor a sua base de conhecimento. Dessa forma, a internet possui uma grande quantidade de informações que cresce cada dia mais, porém, a maioria dessas informações são difíceis de extrair das páginas devido à falta de estrutura de seu conteúdo, porém, ainda assim existem diversas páginas bem estruturadas que em conjunto com técnicas de extração de informações da web, é uma rica fonte de informações.

Para extração automática das informações presentes nas páginas da internet são utilizados robôs baseados em *software*. Robôs são dispositivos que realizam tarefas repetitivas, similar a uma linha de montagem, dessa forma, qualquer programa que realiza um acesso à internet e extrai informações pode ser chamado de robô.

Existem algumas diferentes maneiras de se aplicar um robô na leitura de informações das páginas, as principais são através das técnicas de *spidering* e *web scraping*. Além disso, é possível buscar informações presentes em bancos de dados hospedados em servidor na internet ou outra aplicação web através de chamadas a uma Web API.

2. Web API

Softwares armazenados em nuvem, como plataformas de visão computacional, possibilitam o acesso às suas funcionalidades através da utilização de API. A Web API ganhou força através dos chamados *web services*, este que pode ser definido como qualquer serviço disponibilizado na internet que utiliza troca de mensagens padronizada, geralmente através do XML, para realizar a comunicação sem a presença de uma linguagem ou sistema operacional.

3. Inteligência artificial aplicada ao reconhecimento de imagens

A Visão Computacional é uma subárea que se deriva da Inteligência Artificial, ela tem o objetivo de prover ao computador a habilidade de descrever o mundo que é visto em uma ou mais imagens e reconstruir as suas propriedades, como formas, iluminação, e distribuição de cores.

Alguns dos conceitos utilizados para a composição de uma plataforma de visão computacional são os sistemas de classificação de imagens, estes são compostos de funções matemáticas e alguns conceitos da computação, porém, quando aplicados corretamente, podem servir como base para um complexo sistema de visão computacional, como por exemplo, a categorização automática de fotos nas galerias dos celulares iPhone mais recentes. Dessa forma, existem algumas maneiras de se realizar a classificação de imagens, uma delas é a partir das Redes Neurais.

3.1. Redes Neurais

As Redes Neurais Artificiais são modelos computacionais que possuem similaridades com o que pode ser entender hoje nos estudos de anatomia como sistema nervoso do ser humano. Dessa forma, assim como no sistema nervoso humano, a rede neural toma como sua unidade mínima e base para a suas operações o neurônio.

Inspirado pelo neurônio biológico, um neurônio artificial é composto por:

- a) sinais de entrada: variável utilizada para cálculo do estado;
- b) conjunto de pesos: estes pesos descrevem a força da conexão do neurônio para com outros neurônios que este possui relacionamento;
- c) ativação ou estado: valor produzido pelo cálculo da função de ativação;
- d) função de ativação: função matemática responsável por produzir o resultado que indica a ativação do neurônio.

Cada neurônio deverá possuir uma função de ativação, esta que irá definir o estado do neurônio, chamado de ativação. Para gerar a ativação através do cálculo obtido pela função, o neurônio deverá receber um ou mais valores de entrada, este que poderá ter origem pela ativação de outros neurônios da rede, de camadas anteriores a qual este possui relacionamento, ou de origem externa através entrada da rede neural.

Uma das grandes capacidades das redes neurais que as tornam tão atrativas é a possibilidade de aprendizado baseando-se em um conjunto de amostras apresentadas. Dessa forma, após o treinamento com uma quantidade significativa de amostras, a rede neural é capaz de produzir uma saída que condiz com os exemplos apresentados

3.2 Redes Neurais Convolucionais

Redes neurais convolucionais, do inglês *convolutional neural networks* (CNN), são redes neurais especializadas em processamento de dados em grade, ou matriz, sejam essas grades de uma ou mais dimensões. A extração de características é realizada com a operação de convolução, obtida através da utilização de um filtro, também conhecido como *kernel* ou detector de características, este que irá avaliar os valores da grade de entrada e gerar um mapa da característica sendo avaliada

4. Reconhecimento Ótico de Caracteres

O Reconhecimento Ótico de Caracteres, surgiu para auxiliar a interpretação automática de caracteres em meios impressos. A evolução da tecnologia permitiu avanços da forma como o reconhecimento dos caracteres é realizado. Aproximadamente nos anos 60 a 70 as OCR começaram a ser utilizadas pelos sistemas de correspondências para aumentar a

velocidade da organização das correspondências, esse sistema utilizava um scanner que lia os pontos pretos e brancos da área de endereço de uma carta. A evolução da inteligência artificial nos últimos anos permitiu uma evolução significativa na quantidade de aplicações para uma OCR, isso porque atualmente a maioria das OCR utilizam de *machine learning* e *data mining*.

Dessa forma, diversas plataformas comerciais de visão computacional surgiram nos últimos anos para suprir as novas necessidades de inteligência artificial para reconhecimento de imagens, entre elas, IBM Watson da IBM, Windows Azure da Microsoft, Amazon Rekognition da Amazon e Google Cloud Vision da Google.

4.1 IBM Watson

IBM Watson é uma plataforma comercial em nuvem da IBM, ela visa disponibilizar produtos baseados em inovações de *machine learning* para seus mais variados clientes aplicarem em suas diferentes necessidades. Por ser uma plataforma em nuvem, a integração com os produtos dos clientes é realizada através de *web services*, permitindo a escalabilidade dos recursos conforme o necessário, pois a plataforma é dividida em vários produtos, dessa forma, eles podem ser adquiridos em demanda.

O produto para visão computacional do IBM Watson chama-se Visual Recognition, ele possui modelos de redes neurais treinados para realizar a extração de cenas, objetos, faces, cores, comidas e outras possíveis características de uma imagem. Além disso, é possível disponibilizar amostras para um treinamento personalizado, de forma que o cliente da IBM possa obter os resultados para um campo de conhecimento específico. Dessa forma, o reconhecimento de texto de imagens está disponível através do modelo de detecção pré treinado chamado *Text Model*, do português, Modelo de Texto.

4.2 Windows Azure

A Windows Azure é uma plataforma em nuvem da Microsoft de classificação empresarial que possui soluções para as mais variadas necessidades de uma empresa, desde a hospedagem de aplicações web e ferramentas para desenvolvimento de sistemas em nuvem até a serviços avançados de inteligência artificial baseados em redes neurais.

Computer Vision é um dos serviços disponibilizados na plataforma Azure para a realização de Análise de imagens, podendo assim a partir da análise extrair informações como descrições de objetos da cena, detecção de *tags*, identificar se possui conteúdo adulto, realizar o reconhecimento facial, entre outras coisas. Outro grande aspecto do serviço é o OCR, onde além de extrair das imagens os caracteres impressos, também é possível identificar escrita à mão.

4.3 Amazon Rekognition

Amazon Rekognition é um serviço em nuvem da Amazon para visão computacional, este disponibiliza API de análise de imagens e vídeos para serem utilizados em aplicativos de terceiros. Através do serviço, é possível analisar objetos, pessoas, texto, cenas e atividades, ou também verificar se possui conteúdo não adequado. Além das funcionalidades citadas, é possível realizar também o reconhecimento e comparação facial.

O recurso de API *DetectText* permite realizar a extração do texto de imagens em png e jpg, bastando enviar a imagem através da API para receber uma lista com as palavras detectadas e suas especificações, como confiança, texto detectado, posição na imagem, etc.

4.4 Google Cloud Vision

O Google Cloud Vision é uma API da Google que possui modelos treinados ou permite realizar o treinamento de modelos para realizar análise de imagens. Através da API REST é possível categorizar as imagens, detectar objetos e rostos e também extrair os textos impressos das imagens.

Para realização de treinamento de novos modelos de qualidade, é disponibilizado o recurso '*AutoML Vision Beta*', este que pode ser utilizado até por desenvolvedores com pouco conhecimento em treinamento de redes neurais. Dessa forma, para o treinamento, basta realizar o envio das imagens e aplicar os marcadores desejados, após, o '*AutoML Vision*' treinará um novo modelo que poderá ser escalonado conforme a necessidade para se adaptar a demanda da interface.

5. Implementação de protótipo baseado na tecnologia OCR aplicada ao reconhecimento de rótulos para busca em banco de dados

O trabalho realizado tem como objetivo um protótipo de aplicativo web que permite o cadastro de informações de produtos junto com uma foto do rótulo, desta forma, o rótulo poderia ser utilizado futuramente para auxiliar o acesso as informações cadastradas. A metodologia proposta em teoria pode ser aplicada a qualquer tipo de produto que possua rótulo com texto, porém, foi definido a utilização de cervejas para o protótipo deste projeto.

O protótipo deve armazenar diversas informações da cerveja desejada, dentre as informações estão cervejaria, ingredientes, estilo de cerveja, copos ideais para utilização, entre outros. Porém, o processo de consulta de informações já armazenadas no banco deve ser facilitado, isso é possível ao utilizar a câmera do dispositivo do usuário para enviar uma foto do produto que o usuário tem em mãos, dessa forma, uma imagem do rótulo é enviada para os servidores do aplicativo para o reconhecimento, caso desejado.

5.1 Metodologia para desenvolvimento da arquitetura e protótipo

Primeiramente, observou-se a necessidade de obter dados de produtos para armazenamento um banco de dados com dados de cervejas, dessa maneira, o protótipo final iria possuir uma quantidade considerável de informações. Assim sendo, foram estudadas maneiras de realizar extração de dados com base nos sites disponíveis na internet.

Em segundo lugar, com o intuito de adquirir os conhecimentos necessários do processo de OCR, foi realizado uma pesquisa bibliográfica focando em tecnologias de visão computacional e extração de caracteres de textos impressos, dessa forma, foi possível obter um bom entendimento de como funcionam tecnologias de OCR baseadas em redes neurais, além disso, foi possível obter conhecimento para o desenvolvimento de uma CNN utilizada posteriormente.

Em terceiro lugar, após um bom entendimento das tecnologias de visão computacional, foram pesquisadas algumas informações sobre plataformas comerciais de visão computacional para a utilização no trabalho, visto que não será desenvolvido um mecanismo de OCR a partir do zero.

Em quarto lugar, foi desenvolvido uma pesquisa bibliográfica de alguns dos conceitos norteadores do projeto do protótipo. Dessa forma, uma pesquisa sobre WEB API foi realizada, pois, por exemplo, umas das funcionalidades do protótipo seria uma interface em nuvem para ele possa ser utilizado em qualquer dispositivo com acesso à internet.

Dessa forma, para o desenvolvimento do protótipo, foram definidas e executadas as etapas abaixo:

- a) verificou-se maneiras alternativas ao OCR para reconhecimento de rótulos, como por exemplo, classificadores.
- b) foram realizados testes nas plataformas de visão computacional para ser verificada qual a com melhor precisão para realização de OCR em rótulos de produtos.
- c) foi desenvolvido um núcleo do protótipo, este que permite acesso as suas funcionalidades via API REST
- d) criada uma interface gráfica para a aplicação através do desenvolvimento de um aplicativo com tecnologias da WEB.
- e) foram obtidas informações cadastrais sobre cervejas para inclusão no banco de dados através dos métodos de extração estudados.
- f) foi realizada uma análise das limitações do protótipo obtido e possíveis melhorias.

5.2 Protótipo obtido

Após o desenvolvimento da aplicação, obtivesse uma interface com tela de login, cadastro de usuário e catalogo de cervejas. Após o login, é demonstrado para o usuário o catalogo de cervejas, o usuário pode optar em realizar a consulta de cervejas por texto utilizando o nome da cerveja, estilo, categorias, entre outros, porém, para facilitar a identificação, é possível utilizar também o mecanismo de análise e previsão através da câmera do dispositivo, conforme demonstrado na figura 1.

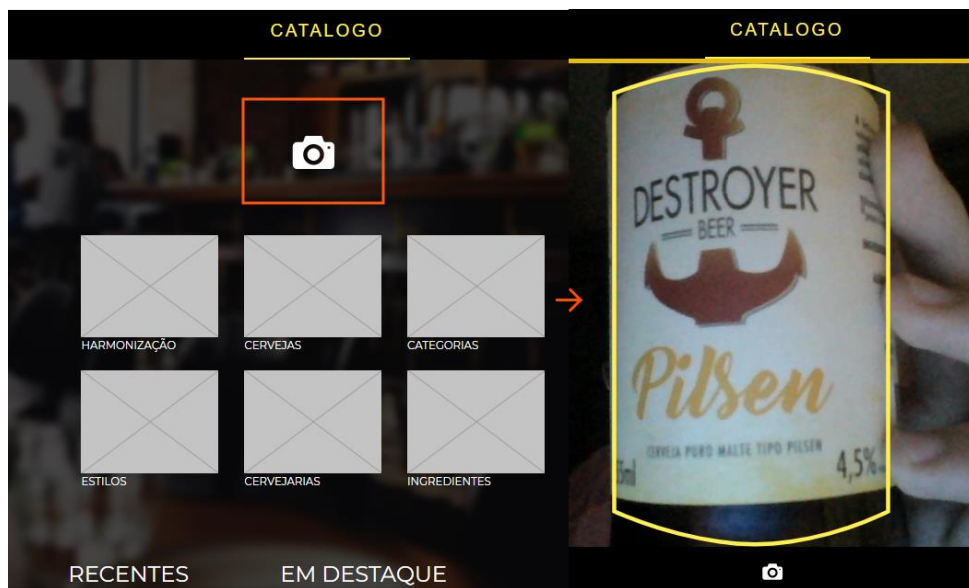


Figura 1. Tela de catalogo e reconhecimento de rótulo do protótipo

Assim que a cerveja desejada ser encontrada, um cadastro detalhado é demonstrado para o usuário, este que ainda pode avaliar a cerveja e comentar a sua avaliação, conforme a figura 2.



Figura 2. Tela de cadastro de cerveja do protótipo

5.3 Resultados

Caso a inclusão de novas cervejas no banco de dados ser realizado por o consumidor do aplicativo, a dependência em exemplares por parte dos classificadores em CNN se torna uma barreira para o mecanismo de detecção de rótulos ser implementado, isso devido a

necessidade de o usuário incluir diversas fotos do mesmo rótulo, que tornaria dificultosa a utilização do aplicativo comercialmente.

Além disso, caso ser optada pela utilização de uma plataforma de visão computacional para realização do OCR para posterior detecção, em testes podemos observar, conforme demonstrado no gráfico de percentual de precisão da figura 3 que, dentre todas as plataformas, a IBM Watson foi a que menos obteve precisão no OCR, seguido por Google Cloud Vision. Dessa forma, é sugerida a utilização das plataformas Amazon Rekognition ou Microsoft Azure.

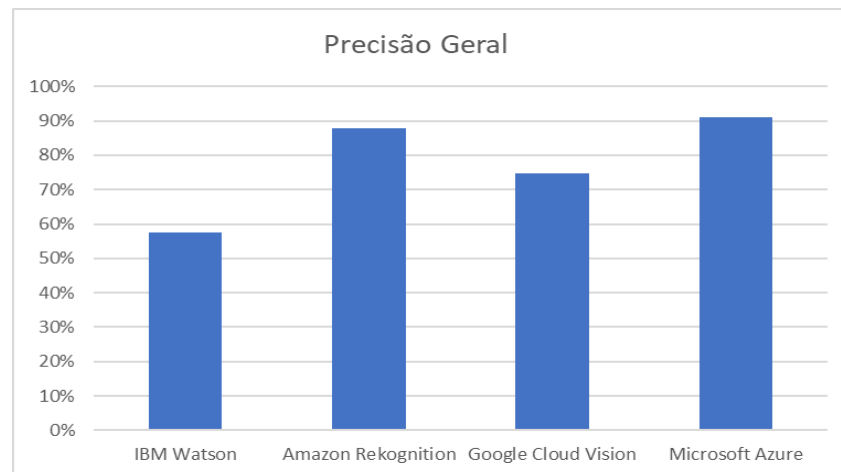


Figura 3. Gráfico de precisão geral na comparação de plataforma de CV

Dessa forma, foi optada a utilização da plataformas de visão computacional Amazon Rekognition para a implantação do mecanismo de reconhecimento devido a facilidades para um período de teste. Assim, foi realizado um teste de precisão com o mecanismo de detecção obtido, o teste contou com apenas três cervejas incluídas no banco, foram realizados três testes de reconhecimentos com cada uma, onde entre todos os testes, em somente uma das análises houve um retorno incorreto. Assim, pode ser concluído que, com poucas cervejas no banco e com um algoritmo simples no mecanismo de análise, o protótipo já apresenta uma precisão de aceitável de 88%.

Além disso, foi possível obter dados completos de cerca de 2500 cervejas com seus respectivos rótulos através de chamadas a Web API de um banco de dados disponível em nuvem, dados que foram armazenados em arquivos, permitindo a inclusão no banco caso desenvolvido um script SQL.

Através das pesquisas e desenvolvimento do protótipo, foi possível identificar algumas das possíveis arquiteturas que um sistema de detecção e detalhamento de rótulos pode ter, além de implementar um protótipo funcional utilizando requisições para realização de OCR em plataformas de visão computacional.

References

- Amazon. Amazon Rekognition. [2018]. Disponível Em: <https://aws.amazon.com/pt/rekognition/>. Acesso Em: 27 Nov. 2018.
- Arasu, Arvind; Garcia-Molina, Hector. Extracting Structured Data From Web Pages. Proceedings Of The 2003 Acm Sigmod International Conference On

- Management Of Data - Sigmod '03, [S.L.], P.337-348, 2003. Acm Press.
<http://dx.doi.org/10.1145/872757.872799>. Disponível Em:
<https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/extract.pdf>.
Acesso Em: 14 Out. 2018.
- Google. Cloud Vision. [2018]. Disponível Em: <https://cloud.google.com/vision/>. Acesso Em: 27 Nov. 2018.
- Heaton, Jeff. Programming Spiders, Bots And Aggregators In Java. Nova York: John Wiley & Sons Inc, 2002.
- Luger, George F. Inteligência Artificial. 6. Ed. São Paulo: Pearson Education Do Brasil Ltda., 2013. 614 P. Tradução De Daniel Vieira.
- Microsoft. Computer Vision. Disponível Em: <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>. Acesso Em: 04 Nov. 2018.
- Neto, Luiz Garcia Palma; Nicoletti, Maria Do Carmo. Introdução Às Redes Neurais Construtivas. São Carlos: Edufscar, 2005.
- Reading And Sorting Mail Automatically. [S.I]: Us Postal Service, 1970. Son., Color. Disponível Em: <https://www.youtube.com/watch?v=cdfyrprhqj0>. Acesso Em: 15 Set. 2018.
- Szeliski, Richard. Computer Vision: Algorithms And Applications. Nova York: Springer, 2011.

ANEXO(S)

ANEXO A - IMPLEMENTAÇÃO DE FRAMEWORK PARA WEB SCRAPING

```

1.  append('http://www.kickstarter.com/discover/recently-launched?page=1')
2.  append('http://www.kickstarter.com/discover/recently-launched?page=2')
3.  wget xpathscrap('//div[@class="project-card"]/*')
4.  branch(: BRANCH_DUPLICATED, : COLLAPSED) {
5.      pipe {
6.          xpath('//h2/strong/a/text()')
7.          decorate(
8.              : head => '\n\t\t<item>\n\t\t\t<title>',
9.              : tail => '</title>'
10.         )
11.         replace(: sourceRE => '&', : dest => "&")
12.         replace(: sourceRE => "'", :dest => "'")
13.         replace(: sourceRE => '"', : dest => '"')
14.     }
15.     pipe {
16.         xpath('//p/text()')
17.         decorate(
18.             : head => '\n\t\t\t<description>\n\t\t\t\t\t',
19.             : tail => '\n\t\t\t\t</description>'
20.         )
21.         replace(: sourceRE => '&', : dest => "&")
22.         replace(: sourceRE => "'", :dest => "'")
23.         replace(: sourceRE => '"', : dest => '"')
24.     }
25.     pipe {
26.         xpath('//div[@class="Project-thumbnail"]/a/img/@src')
27.         decorate(
28.             : head => '\n\t\t\t<image>\n\t\t\t\t\t<url>',
29.             : tail => '</url>'
30.         )
31.     }
32.     pipe {
33.         xpath('//h2/strong/a/text()')
34.         decorate(
35.             : head => '\n\t\t\t\t\t<title>',
36.             : tail => '</title>\n\t\t\t\t<link>http://www.kickstarter.com</link>\n\t\t\t\t</image>'
37.         )
38.     }
39.     pipe {
40.         xpath('//div[@class="project-thumbnail"]/a/@href')
41.         decorate(
42.             : head => '\n\t\t\t\t<link>',
43.             : tail => '</link>\n\t\t\t</item>'
44.         )
45.     }
46. }
47. decorate(
48.     : head => '<?xml version="1.0" encoding="UTF-
49.     8"?>\n<rss version="2.0">\n\t<channel>\n\t\t<title>Kick Starter (Recently Launched)</title>
50.     \n\t\t<link>http://www.kickstarter.com</link>',
51.     : tail => '\n\t</channel>\n</rss>'
52. )

```

Fonte: GITHUB.COM/MICHADA (2018).

ANEXO B – MODELO JSON DA CERVEJA OBTIDO DE BREWERYDB.COM

```

{
  "id": "4YeNYL", // Id da cereveja
  "name": "1635 English Mild", // Nome da cerveja
  "abv": "4.2", // Percentual alcoolico
  "ibu": "20", // Amargor
  "glasswareId": 5, // Id do copo recomendado
  "styleId": 10, // Id do estilo da cerveja
  "isOrganic": "N", // Organica
  "glass": { "id": 5, "name": "Pint", "createDate": "2012-01-03 02:41:33" },
  // Objeto do copo recomendado
  "style": {
    "id": 10, // Id do estilo
    "categoryId": 1, // Id da categoria do estilo
    "category": {
      "id": 1, // Id da categoria do estilo
      "name": "British Origin Ales" // Nome da categoria
    }, // Objeto da categoria do estilo da cerveja
    "name": "English-Style Pale Mild Ale", // Nome do estilo
    "description": "English pale mild ales range from golden to amber in
color. Malt flavor dominates the flavor profile with little hop bitterness or
flavor. Hop aroma can be light. Very low diacetyl flavors may be appropriate
in this low-alcohol beer. Fruity-ester level is very low. Chill haze is
allowable at cold temperatures." // Descrição do estilo
  }, // Objeto do estilo da cerveja
  "breweries": [
    {
      "id": "Yf5yWv", // Id da cervejaria
      "name": "Iechyd Da Brewing Company", // Nome da cervejaria
      "images": {
        "squareLarge":
"https://s3.amazonaws.com/brewerydbapi/brewery/Yf5yWv/upload\_VAdala-squareLarge.png" // Logo da cervejaria
      },
      "locations": [
        {
          "id": "CpYuuR", // Id da cervejaria
          "name": "Main Brewery" // Nome da cervejaria
        }
      ]
    }
  ] // Lista de objetos das cervejarias responsáveis pela cerveja
}

```